

A. WEILER — J. SCHIEB

MICRO APPLICATION

12

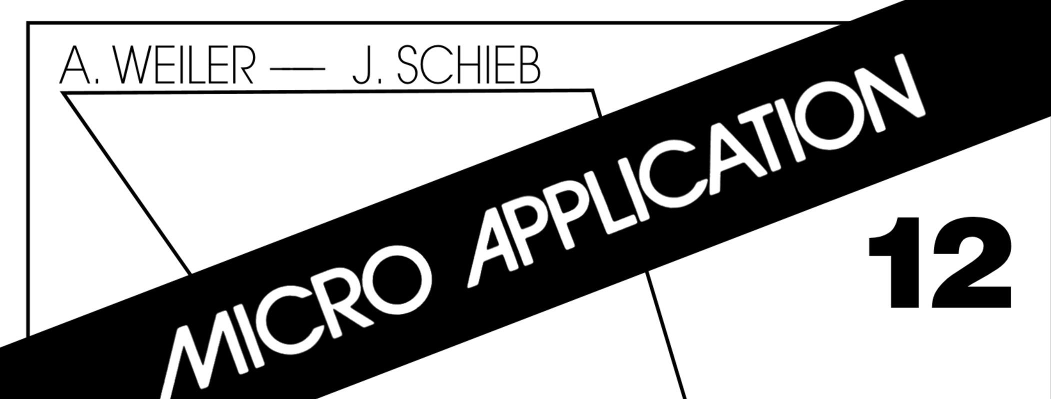
AMSTRAD

LE LIVRE DU CP/M2.2 ET CP/M PLUS (3.0)

pour les CPC 464, 664 et 6128



UN LIVRE DATA BECKER



AMSTRAD

LE LIVRE DU CP/M2.2 ET CP/M PLUS (3.0)

pour les CPC 464, 664 et 6128



UN LIVRE DATA BECKER

Distribué par : MICRO APPLICATION
13, Rue Sainte Cécile
75009 PARIS

et

EDITION RADIO
3, Rue de l'Eperon
75006 PARIS

ISBN : 2-86899-025-8

Traduction Française assurée par Pascal Hausman

édité par Frédérique BEAUDONNET

TABLE DES MATIERES

I. L'ordinateur

Pour vous dire la vérité, je dois reconnaître que je vous envie car vous savez déjà ce que vous ne savez pas. Moi, par contre, je ne peux qu'essayer de m'imaginer jusqu'où vont vos connaissances sur les ordinateurs et tout ce qui s'y rapporte. Comme il ne m'est malheureusement pas possible d'interroger chacun d'entre vous, je vais donc commencer pour ainsi dire dans ce chapitre par "Adam et Eve", de façon à ce que tous aient la même base de départ et que nous puissions ainsi nous comprendre par la suite. On rencontre partout des façons semblables de procéder. Voyez par exemple les normes techniques ou essayez d'écrire un programme pour votre ordinateur, vous verrez que les hommes ont toujours recours à des conventions bien définies, de façon à bien se comprendre.

Eh bien, allons-y. Tout d'abord, qu'est-ce qu'un ordinateur? En suivant une stricte méthode scientifique, nous allons maintenant expliquer ce qu'est un ordinateur.

Nous allons tout d'abord nous faire aussi bête que possible et dire: "Un ordinateur est une boîte dans laquelle quelque chose se passe". Heureusement le nom anglais de l'ordinateur, "Computer" nous donne déjà une idée de ce qui se passe justement dans un ordinateur. "Compute" signifie "compter" (les deux verbes ont d'ailleurs la même origine) et l'ordinateur est donc un calculateur. Un tel jongleur avec les chiffres convient parfaitement pour traiter des masses de chiffres importantes pour la gestion ou pour l'évaluation de séries statistiques. Les écoliers et les étudiants n'auront certainement pas de mal à trouver d'autres applications pour un tel esclave de calcul.

I.1 Le clavier

Mais il y a un problème. Nous avons simplement devant nous une boîte capable d'effectuer des calculs. Mais comment lui indiquerons-nous ce qu'elle doit calculer? Une première possibilité serait de lui murmurer quelque chose à l'oreille. Mais ce mode de communication avec les ordinateurs appartient pour le moment encore au domaine de la science fiction et nous ne pouvons l'utiliser. C'est pourquoi nous pouvons chercher à employer ce bon vieux clavier des machines à écrire. Nous connecterons donc un tel "appareil d'entrée de lettres et de caractères" à notre boîte.

Mais comme l'ordinateur est beaucoup plus puissant qu'une machine à écrire, le clavier devra également se présenter un peu différemment. Un clavier d'ordinateur dispose d'entre 60 et 100 touches, suivant sa classe de prix et sa facilité d'utilisation. Il y a souvent, à côté d'un clavier principal bien ordonné avec les lettres, les chiffres et les caractères, un petit clavier séparé pour l'entrée rapide de colonnes de chiffres et même, sur les ordinateurs les plus pratiques, encore un clavier spécial avec ce qu'on appelle les touches de fonction.

Vous pouvez facilement voir si votre ordinateur possède un clavier français ou américain. Si vous lisez dans la ligne supérieure des lettres, en partant de la gauche, le mot AZERTY, vous êtes l'heureux propriétaire d'une unité d'entrée correspondant aux habitudes françaises. Si vous lisez par contre à cet endroit, QWERTY, vous pourrez utiliser certainement plus facilement les nombreux programmes informatiques qui nous viennent des Etats-Unis, mais vous devez renoncer aux jolies lettres "àèéùç", etc. On ne peut pas tout avoir.

Quel que soit le clavier, français ou américain, dont vous disposez, il y a un certain nombre de touches qui sont très importantes en informatique et qui ont souvent les mêmes noms sur les deux types de clavier. Comme ces exceptions sont importantes pour le reste de cet ouvrage et pour tous les programmes, nous allons maintenant vous indiquer de quelles touches il s'agit:

RETOUR DE CHARIOT

Vous connaissez déjà cette touche pour les machines à écrire mais elle est le plus souvent utilisée de façon totalement différente sur les ordinateurs. Les inscriptions sur cette touche se présentent souvent ainsi:

RETURN (=retour de chariot)

ENTER (=fin d'une entrée)

CR ("Carriage Return" = retour de chariot)

<— (symbole graphique du retour de chariot)

PAS EN ARRIERE

BS (=Back Step)

TOUCHE DE SUPPRESSION OU DE CORRECTION

DELETE (=supprimer)

DEL (abréviation du précédent)

RUB OUT (même signification)

MAJUSCULES

SHIFT (=décaler -sous-entendu: sur majuscules)

FIXER SUR MAJUSCULES

SHIFT LOCK

La fonction exacte de cette dernière touche dépend de la construction de votre clavier. Parfois ce n'est pas l'ensemble du clavier qui est décalé mais seulement les lettres majuscules. Cela peut représenter un allègement considérable de votre travail. Certains claviers possèdent en outre des touches spéciales pour le décalage des lettres:

ALPHA LOCK ou

CAPS LOCK

Et si vous avez vraiment de la chance, l'état actuel du clavier est indiqué par une loupiote.

Une touche très importante en informatique est la touche CONTROL qui se trouve sur la droite de votre clavier AMSTRAD. Cette touche doit être appuyée en liaison avec une lettre ou un chiffre pour de nombreuses instructions de commande. Elle porte en général l'inscription:

CTRL ou (sur l'AMSTRAD)

CTL

I.2 L'écran

Maintenant que nous avons la possibilité d'entrer des données dans la boîte, nous avons bien sûr également besoin d'un moyen pour que les données puissent nous parvenir à nouveau. Cela ne nous serait en effet d'acune utilité si l'ordinateur pouvait faire les calculs les plus fantastiques mais qu'il ne puisse pas nous communiquer ce qu'il a obtenu. La sortie des données sur un écran est un moyen très pratique car cela fonctionne très vite et cela ne fait pas de bruit. Les données et les messages d'erreur de l'ordinateur sont rapidement visibles et si l'on veut avoir une trace écrite de ses erreurs, on peut toujours faire entrer une imprimante en action.

Un écran est au fond un téléviseur "reconverti" qui fonctionne d'après le même principe. Seulement, sur cette fenêtre de l'ordinateur qu'on appelle souvent aussi moniteur, ce ne sont pas le plus souvent des images en défilement qui sont représentées, mais des lettres, des chiffres et autres caractères. C'est pourquoi les exigences qu'on doit avoir pour un moniteur sont différentes de celles qu'on aura pour un téléviseur. La résolution devrait être nettement meilleure pour qu'on ne se fatigue pas les yeux lorsqu'on reste plusieurs heures devant son moniteur.

I.3 L'imprimante

A l'époque préhistorique des gros ordinateurs peu puissants aux regards de nos critères actuels, les imprimantes étaient utilisées comme moyen de sortie des données sous une forme durable. L'ordinateur envoyait donc ses données à l'imprimante en gâchant en règle générale une grande quantité de papier. Outre la grande consommation de papier, une autre inconvénient est constitué par le bruit que font les imprimantes pour écrire les caractères sur le papier. Mais beaucoup de gens tiennent encore à avoir des informations sous une forme écrite et l'on ne peut pas faire l'économie d'une imprimante. Comme il y en a beaucoup et de très différentes, voici un petit aperçu:

Les prix et les possibilités sont extrêmement variables. C'est ainsi que les imprimantes peuvent coûter entre 1500 et 30 000 francs et qu'elles offrent des vitesses d'écriture de 15 à 800 caractères par seconde.

I.3.1 Les imprimantes à aiguille

Les imprimantes à aiguille sont bien adaptées au travail avec les microordinateurs. Elles occupent la première place sur le marché et c'est de leurs rangs que viennent aussi bien les imprimantes les moins chères que les plus rapides. Tout a commencé avec les imprimantes à aiguille qui fixent bien proprement un point à côté d'un point, produisant ainsi une écriture qui n'était pas propre du tout. Cela tient au principe de l'impression des temps "préhistoriques" (vers 1980). Des aiguilles commandées individuellement frappent un ruban encreur puis le papier, représentant ainsi un petit point. Avec sept points verticalement, on peut déjà reconnaître une lettre ou un chiffre. Mais la lecture de longs textes est vraiment fastidieuse.

Mais ces temps sont révolus. Si l'on excepte les machines les moins chères, les imprimantes à aiguille offrent aujourd'hui une largeur d'écriture de 80 caractères, une vitesse autour de 100 caractères par seconde et une forme de caractères qui ne convient peut-être pas pour la correspondance mais qui peut être employée pour des écrits secondaires. Le grand avantage des imprimantes à aiguille: elles offrent en général un grand nombre d'écritures ainsi que des jeux de caractères que l'on peut charger ou programmer à volonté et que l'on peut très facilement sélectionner. C'est ainsi que la correspondance avec la Grèce ou le Japon est considérablement facilitée.

Les imprimantes les plus puissantes atteignent des vitesses allant jusqu'à 800 caractères par seconde avec une largeur d'écriture de 132 caractères mais elles coûtent environ entre 25 000 et 30 000 francs. Relativement nouvelles sont les imprimantes du "nouveau type".

Elles possèdent le plus souvent 24 aiguilles d'impression et peuvent ainsi, à une vitesse moindre, atteindre la qualité des machines à écrire. Les performances les plus courantes sont ici d'environ 70 caractères par seconde pour la correspondance et de 140 à 150 caractères en mode d'écriture rapide.

I.3.2 Les imprimantes à marguerite

Une seconde catégorie est constituée par les imprimantes à marguerite et les imprimantes à boule. Ces dernières sont en général basées sur le principe de la construction des anciennes machines à écrire et leur entretien est souvent coûteux. D'autre part, leur fabrication coûte cher. Les imprimantes à marguerite ont remplacé la boule d'impression par un disque portant les différentes lettres et caractères. Un petit marteau frappe les caractères sur le papier et on ne peut pas voir de différence par rapport à une machine à écrire. Pour une correspondance soignée, l'imprimante à marguerite convient donc parfaitement.

Vitesse: sur les modèles bon marché, le plus souvent 20 caractères par seconde et 80 caractères par seconde pour les modèles de pointe.

I.3.3 Les imprimantes à jet d'encre

Les nouvelles imprimantes à jet d'encre sont nettement meilleur marché. Comme l'imprimante à aiguille, elles créent les lettres et les caractères à partir de combinaisons de points. Mais il n'y a pas pour ce faire de contact mécanique entre la tête d'impression et le papier mais de minuscules gouttes d'encre sont bombardées sur le papier. La qualité d'écriture est comparable à celle des imprimantes à aiguille simples. Avantages principaux des imprimantes à jet d'encre: très faible niveau de bruit d'environ 45 dba (décibels A), pour une vitesse atteignant cependant 150 caractères par seconde. Inconvénient: les copies ne peuvent pas

être effectuées avec des feuilles de carbone mais uniquement par des impressions répétées.

I.3.4 Imprimantes thermiques

Les imprimantes thermiques présentent le même inconvénient. Elles produisent les caractères en agissant sur la température de la surface du papier. Naturellement il faut utiliser pour cela un papier spécial qui jaunit à la longue à la lumière. Cependant ces imprimantes, qui travaillent également d'après le principe des combinaisons de points, sont très bon marché et sont tout à fait adaptées pour des fonctions de contrôle immédiat. Les imprimantes thermiques sont déjà utilisées sur les calculatrices de la classe de prix inférieure à 300 francs.

L'idéal pour une impression rapide et propre avec un ordinateur personnel est représenté par les imprimantes à laser. L'écriture produite ne se distingue pas d'une impression professionnelle par composition.

Mais je pense soudain à quelque chose. Notre boîte, qui ressemble, par la description que nous en faisons, de plus en plus à un ordinateur, doit bien sûr comporter un commutateur qui nous permette de couper l'arrivée du courant. Avec l'intention louable de rendre le plus difficile possible l'allumage ou l'arrêt par erreur, de nombreux fabricants d'ordinateurs ont pensé que le commutateur doit toujours être relativement caché. Si vous n'avez pas encore trouvé le commutateur de votre ordinateur, cherchez-le à un endroit où personne n'aurait l'idée de le chercher. C'est certainement là qu'il se trouve.

I.4 Périphériques de stockage des données

Notre ordinateur théorique nous permet déjà de faire pas mal de choses. L'ordinateur lui-même est là et nous avons déjà également un écran et une imrimante. Ce qui nous manque, c'est un endroit pour stocker les données. Nous devons pouvoir stocker les données d'abord dans l'ordinateur mais aussi -et pour plus longtemps- en dehors de l'ordinateur. On peut dire que, dans le principe, la façon de stocker des données est identique quel que soit le moyen qu'utilise l'ordinateur. Que ce soit ce qu'on appelle la mémoire de travail, c'est-à-dire celle qui conserve le texte que je suis en train d'écrire, ou que ce soit une disquette ou un disque dur. Comment l'ordinateur peut stocker et comment il stocke des données, c'est ce que nous allons voir maintenant.

Comme il s'agit de notions un peu théoriques, vous pourriez peut-être aller vous chercher une tasse de café, en boire une bonne rasade avant de reprendre attentivement votre lecture. J'attends que vous reveniez.

I.5 Un ou Zéro

Vous savez bien sûr que les ordinateurs sont des appareils électriques. Par conséquent les informations doivent également être traitées à l'intérieur de l'ordinateur sous une forme électrique. Mais comment y parvenir? Comment pouvons-nous indiquer à un ordinateur ce que nous pensons? Il faut pour cela un mode de représentation que l'ordinateur comprenne. Il y a un mode de représentation que l'ordinateur connaît: les deux états: courant passe - aucun courant ne passe. C'est d'après le même principe que fonctionne une ampoule électrique. Si elle remarque que le courant passe, elle commence vite à brûler. Si elle s'aperçoit qu'il n'y a plus de courant, elle s'éteint. Ce qui est à la portée d'une stupide ampoule électrique est bien sûr très facile pour notre

ordinateur. Ses constructeurs ont écrit dans son cerveau d'électrons que l'état "courant passe" correspond à la valeur "Un". De même que le "un" a pour nous une valeur bien déterminée.

Si aucun courant ne passe, cela signifie pour l'ordinateur: "0". Il peut donc ainsi distinguer maintenant entre deux états différents, ce qui conduit à des résultats étonnants.

1.6 Compter en binaire

Pour un usage quotidien, compter en binaire, c'est-à-dire avec des 0 et des 1, n'est pas particulièrement adapté. Avant que vous n'arriviez à indiquer votre date de naissance sous cette forme, vous aurez déjà la langue sèche. Heureusement, l'ordinateur n'a pas de langue et rien ne risque donc de s'assécher lorsqu'il jongle comme un fou avec ces deux chiffres.

Examinons un peu le principe du travail avec des nombres. Il existe différents systèmes numériques: binaire, décimal, hexadécimal et quelques autres. Le système décimal est bien sûr celui que nous maîtrisons tous. Il nous est très difficile de penser autrement qu'en valeurs décimales. Mais que faisons-nous donc alors? Nous pensons à un nombre, par exemple le zéro. S'il s'agit d'une valeur plus importante, nous prenons le un, et cetera. Une fois arrivé au "9", nous constituons le prochain nombre à partir des deux plus petits, ce qui nous donne un "10".

Nous pouvons procéder de la même manière lorsque nous n'avons que deux chiffres à notre disposition. La première valeur est le zéro. La prochaine le un. Après c'est fini, nous n'avons plus d'autre chiffre pour compter plus loin. Nous constituons alors le prochain nombre avec ces deux chiffres et nous obtenons 10 pour la valeur 2. Le trois se présente ainsi: 11 et pour le quatre nous avons déjà à nouveau besoin d'un chiffre supplémentaire, ce qui nous donne 100.

Pour bien comprendre, voici ici encore une fois une table des nombres binaires:

Zéro = 0
Un = 1
Deux = 10
Trois = 11
Quatre = 100
Cinq = 101
Six = 110
Sept = 111
Huit = 1000

et ainsi de suite.

Si l'on considère les chiffres de ce système numérique, on peut représenter deux valeurs avec un chiffre, quatres valeurs avec deux chiffres, huit valeurs avec trois chiffres, 16 valeurs avec quatre chiffres, et cetera. Quand le nombre de chiffres augmente de 1, le nombre de possibilités est toujours multiplié par deux. Nous connaissons le même principe pour le système décimal, si ce n'est que tout va alors par dix. Un chiffre peut représenter dix valeurs, deux chiffres cent valeurs, trois chiffres mille...

Cette histoire de nombres avec deux chiffres s'appelle système numérique binaire (de base deux) et a tellement plu à ceux qui ont répandu la mode des hamburgers mais aussi des ordinateurs qu'ils ont tout de suite inventé pour ce système numérique un certain nombre d'expressions techniques.

Un chiffre dans un nombre se dit "digit" en américain et binaire se dit "binary". Les deux ensemble donnent "binary digit", ce qui a été abrégé en "bit".

Mais un seul chiffre représente une place en mémoire vraiment faible et les ordinateurs modernes ont une grande masse de places mémoire. Si l'on raisonnait en bits, les nombres indiquant la place mémoire disponible seraient énormes. C'est pour cette raison qu'on a créé des unités, des "mots" qui désignent par un seul mot ou un seul nombre une grande quantité de bits. On peut ainsi désigner 256 bits avec un mot de huit bits et 65536 avec un mot de 16 bits.

L'unité offrant 256 possibilités est très souvent utilisée et c'est pourquoi on lui a donné le nom de "byte". En français on dit "octet". On raconte que c'est la raison pour laquelle la revue technique américaine "Byte" ne doit jamais paraître avec plus de 256 pages. En réalité elle en comporte plus du double.

I.7 Stocker des valeurs

Pour pouvoir travailler correctement, l'ordinateur doit également pouvoir stocker les nombres. Un chiffre, comme nous l'avons vu, est constitué pour l'ordinateur par un état électrique: chargé ou non-chargé. Si l'on réunit huit de ces places mémoire, on obtient un mot de huit bits qui peut représenter en tout 256 valeurs. Le "hasard" veut que cela corresponde justement à un octet. Mais comme un octet ne représente pas encore beaucoup d'informations, il y a encore le terme de kilo-octets. Mais à cet égard, un kilo ne correspond toutefois pas exactement à 1000 octets mais, du fait du mode de comptage binaire, à la puissance de deux la plus proche, 1024. Pour les très grandes unités existe en outre le méga-octets.

Avec l'aide de branchements électroniques, il est maintenant possible de lire les valeurs d'un octet, de les traiter et de stocker à nouveau de nouvelles valeurs. C'est là tout le secret de l'ordinateur. Son intérêt vient de ce qu'il peut effectuer ces opérations en très peu de temps et qu'il constitue ainsi un véritable et très utile outil de traitement des données.

Comme chaque emplacement de la mémoire doit exister réellement, toute mémoire d'ordinateur ne dispose que d'une place limitée dans laquelle les données peuvent être placées. Les ordres de grandeur usuels pour la mémoire de travail d'un ordinateur personnel qui travaille avec une unité centrale 8 bits, l'unité centrale étant le véritable cerveau de l'ordinateur, sont de 64 kilo-octets. Les nouveaux ordinateurs 16 bits offrent des dimensions de mémoire entre 128 K et 8 méga-octets.

Un point n'est cependant pas encore éclairci jusqu'ici, à savoir

comment il est possible de stocker du texte dans un ordinateur alors que celui-ci ne travaille qu'avec des chiffres.

A cet égard les ordinateurs se comportent vraiment comme les hommes. Ils utilisent, exactement comme nous quand nous ne comprenons pas une langue étrangère, un dictionnaire. A l'intérieur de l'ordinateur, il y a une table dans laquelle une valeur numérique est fixée pour chaque lettre, ainsi que pour tous les autres caractères. Si donc une lettre est entrée dans l'ordinateur, il consulte sa table pour y trouver la valeur binaire correspondante. Il n'aura plus alors de problème pour traiter la valeur numérique ainsi obtenue. Les choses fonctionnent exactement de la même façon pour la sortie.

Pour que ce processus ne fonctionne pas seulement sur un seul ordinateur d'un fabricant déterminé, il faudrait s'accorder sur la "table de traduction". Malheureusement, d'un point de vue français, cet accord s'est fait d'abord aux Etats-Unis et ne concerne donc que les caractères utilisés dans ce pays. Cet accord s'appelle ASCII (American Standard Code for Information Interchange = code standard américain pour les échanges d'informations) et il ne concerne donc pas les accents et caractères spéciaux français. Ce code ASCII comprend 128 caractères qui ont chacun reçu une valeur propre.

Mais que fait l'ordinateur avec une mémoire pleine de valeurs? Rien. Il est trop stupide pour pouvoir tirer quoi que ce soit de ces données. Pour que quelque chose se passe, il faut d'abord que vous lui donniez une instruction adaptée. Ce n'est que si vous lui donnez des instructions appropriées que l'ordinateur peut accomplir quelque chose. Au nouveau le plus élémentaire, le processus se présente ainsi: vous dites à l'ordinateur: prends le bit dans la mémoire "a", ajoute-le au bit dans la mémoire "b" et place le résultat dans la mémoire "c".

Toute une série d'instructions de ce type sont déjà stockées dans la Central Processing Unit (CPU = unité centrale), le cerveau de calcul de l'ordinateur. A partir de beaucoup de ces opérations, on peut reconstituer des procédures beaucoup plus complexes, ce qui

est le but de la programmation en langage machine ou langage assembleur. Ce sont des langages de programmation qui commandent directement les fonctions de la machine. Le Basic, Pascal ou Fortran ne commandent la CPU qu'après une traduction interne en langage machine du programme correspondant.

Pour résoudre une tâche déterminée, chaque étape doit être prescrite à l'ordinateur avec une extraordinaire minutie. Cela s'effectue habituellement à l'aide d'un programme qui n'est pas autre chose qu'une succession d'instructions qui seront traitées les unes après les autres. Le programme lui-même doit bien sûr également se trouver dans la mémoire pour qu'il puisse être exécuté et il prend donc également de la place en mémoire.

Suivant la tâche à réaliser, un tel programme peut occuper entre une dizaine de K et 120 K octets. Il peut donc arriver que la place en mémoire soit déjà entièrement occupée par le programme de sorte qu'il ne reste plus de place pour l'exécution du programme et pour les données de l'utilisateur. Cela signifie alors que ce programme est trop grand pour cet ordinateur et qu'il ne peut être utilisé. Mais ces cas sont vraiment rares. Il arrive plus souvent que les données de l'utilisateur ne rentrent plus dans la mémoire de travail intégrée.

I.8 Mémoire de masse

Il est rarement indispensable que la totalité d'un programme figure à un moment donné en mémoire. Il est le plus souvent tout à fait suffisant que les parties de programme les plus importantes soient en mémoire et que le reste ne soit chargé en mémoire qu'en cas de besoin. C'est pour cette raison que les ordinateurs disposent, outre leur mémoire interne de travail, également de possibilités externes de stockage qui disposent d'une place mémoire moins limitée et qui sont en outre meilleur marché. Ce qu'on appelle les mémoires de masse peuvent être: un lecteur de cassette, une unité de disque souple ou un disque dur. Ces trois moyens ont leurs avantages et leurs inconvénients mais ils offrent tous de la place à profusion pour les données et ils peuvent échanger ces données avec l'ordinateur en un temps acceptable.

I.8.1 Disque souple

Le disque souple ou floppy est une sorte de disque faite à partir de la matière d'une bande magnétique. C'est un disque mince dont la surface est magnétisée et qui est enveloppée dans une enveloppe le protégeant contre les doigts gras de l'utilisateur et les particules de poussière et de saleté. Lorsque la disquette souple est placée dans l'ordinateur, une tête de lecture/écriture se déplace au-dessus de la surface de la disquette et magnétise ou démagnétise certains emplacements.

Nous avons là à nouveau deux états: magnétique ou pas. Cela se recoupe bien sûr volontairement avec la méthode de stockage des données à l'intérieur de l'ordinateur. Les données qui figurent dans l'ordinateur sous forme d'états électriques peuvent ainsi être transférées sur la disquette. Si un bit possède l'état chargé, la tête d'écriture du lecteur de disquette souple magnétisera un petit point de la surface de la disquette par une impulsion électrique. Si le bit suivant n'est pas chargé, il n'y aura rien non plus sur la disquette. A partir d'une suite de points magnétiques et non

magnétiques sur la disquette, il est alors possible de lire un programme ou une série de données et de les transférer à nouveau dans l'ordinateur. Nous avons ainsi la possibilité de stocker des données à l'extérieur de l'ordinateur.

La disquette tourne avec 200 à 300 rotations par minute et on obtiendrait un terrible chaos de données si on écrivait les données au hasard sur la disquette. On subdivise donc la disquette en un nombre déterminé de pistes qui figurent l'une à côté de l'autre sur la disquette, comme les voies d'une autoroute.

Lorsque l'on veut placer des données sur la disquette, la tête d'écriture se déplace vers la piste correspondante et écrit les informations. Mais une telle piste est encore vraiment longue et il faudrait attendre assez longtemps si l'on voulait lire des données à partir d'une telle piste.

Pour diminuer le temps de lecture, on subdivise encore la disquette en secteurs (leur taille varie selon le fabricant de 128 à 1024 octets et elle est sur l'AMSTRAD CPC de 512 octets) qui ressemblent à des parts de gâteau.

Cette méthode d'écriture est très pratique mais elle interdit cependant entre autre qu'on puisse placer simplement la disquette d'un ordinateur dans le lecteur de disquette d'un autre ordinateur pour lire les données avec ce second ordinateur. Chaque fabricant essaie en effet d'imaginer sa propre méthode pour nous ennuyer, nous les utilisateurs. Dans ce domaine les firmes Apple et Victor se distinguent particulièrement.

Lorsque vous achetez des disquettes neuves non formatées, vous devez en règle générale les formater avant de pouvoir les utiliser. Il existe à cet effet sur chaque ordinateur un programme spécial qui écrit précisément sur la disquette les pistes et les secteurs dont l'ordinateur a besoin.

Si vous aimez bien vous énerver, essayez donc un jour de formater une de vos disquettes comportant des informations importantes. Vous serez surpris de voir à quel point il en restera peu de chose. Il

vaut donc mieux que vous fassiez attention et que vous placiez une protection contre l'écriture sur l'entaille de la disquette. Les petits auto-collants prévus à cet effet et qui sont joints à toute boîte de disquettes peuvent vraiment vous sauver la mise. Sur les disquettes trois pouces telles qu'elles sont utilisées sur l'ordinateur AMSTRAD, il suffit d'actionner un bouton en plastique pour empêcher l'ordinateur d'écrire sur une disquette déterminée.

Il est cependant tout aussi prudent et sensé de toujours faire une copie de sécurité de toutes les données importantes et de la conserver à l'abri des enfants, des chiens, des chats, du feu, des cambrioleurs, des tempêtes de neige, des tremblements de terre et autres calamités. Si vous ne placez pas vos diamants dans votre coffre, vous devriez au moins y placer vos copies de sécurité. Vous apprendrez dans un chapitre suivant comment vous pouvez effectuer des copies avec CP/M.

Il existe actuellement sur le marché des formats de disquette très différents. On trouve encore la très respectable disquette 8 pouces qui ouvrit jadis la voie à la marche triomphale du traitement électronique des données. Les disquettes 5 pouces 1/4 sont particulièrement répandues et sont normalement utilisées sur les ordinateurs personnels. Le format est assez maniable et l'on peut placer jusqu'à presque deux méga-octets sur une telle disquette.

Mais, notamment au Japon, les disquettes 3 pouces et 3 pouces et demi commencent à se répandre sur le marché. Elles permettent une manipulation encore améliorée car l'ouverture de lecture/écriture est fermée par une protection métallique tant que la disquette ne se trouve pas dans le lecteur de disquette. Les disquettes trois pouces disposent en outre d'une enveloppe plastique dure de sorte qu'on ne peut plus les plisser qu'en les cassant volontairement.

I.8.2 Le disque dur

Sur les ordinateurs qui sont essentiellement utilisés à des fins professionnelles, on trouve les disques durs. Ces disques durs sont

identiques par leurs dimensions à une disquette 5 pouces 1/4 et ils occupent la même place dans un habitacle d'ordinateur (comme la vague de miniaturisation ne s'arrête devant rien, on trouve bien sûr déjà des disques durs de trois pouces et demi). Mais à l'intérieur, des disques métalliques tournent à 3000 rotations par minute. Une particule de poussière ou de fumée sur la surface d'un disque dur peut avoir des effets désastreux.

C'est pourquoi les disques durs sont placés dans un habitacle hermétique à l'air qu'il ne devraient jamais quitter. Depuis le début des années 80 on peut dire que la capacité standard d'un disque dur est de 10 M octets. Mais les prix baissent et le disque 20 méga a déjà fait son apparition.

Outre son énorme capacité, le disque dur offre encore un avantage décisif: il est jusqu'à 20 fois plus rapide qu'une disquette pour l'accès aux données.

Eh bien si j'ai bien compté, je crois que nous avons maintenant évoqué toutes les parties d'un véritable ordinateur et vous devriez maintenant posséder les connaissances de base sur cet appareil puissant et parfois aussi agréable. Dans le chapitre suivant nous allons vous expliquer ce qu'est un système d'exploitation et à quoi cela sert.

1.9 Ce que vous savez déjà

* Vous connaissez maintenant les différents éléments dont se compose un ordinateur, du clavier à l'écran.

* Vous connaissez les différents types d'imprimantes avec leurs avantages et leurs inconvénients.

* Vous savez également ce qu'est un moyen de stockage.

* Vous connaissez les chiffres un et zéro, vous pouvez compter en binaire et vous savez comment les valeurs sont stockées.

II. Le système d'exploitation

Dans le chapitre précédent nous vous avons donné un aperçu de l'électronique d'un ordinateur et de ses différents éléments. Si vous regardez la valeur matérielle d'un tel ordinateur, vous constaterez que celle-ci n'est pas très élevée. La raison en est qu'avec les composants électroniques d'un ordinateur vous ne pouvez en fait rien faire du tout. Un ordinateur ne commence à avoir vraiment de valeur que lorsque vous pouvez faire tourner sur cet ordinateur des programmes qui sont importants et utiles pour vous. Pour que cela fonctionne, vous avez besoin de ce qu'on appelle le logiciel. Mais ici aussi il nous faut distinguer différents concepts.

Pour minimiser la confusion générale qui règne dans ce domaine, nous nous en tiendrons à deux types de logiciels: le système d'exploitation et les programmes. Il faut cependant que vous gardiez présente à l'esprit l'idée que le système d'exploitation est aussi un programme. La distinction qu'il convient de faire tient donc à la fonction des différents logiciels comme nous allons maintenant vous l'expliquer.

On entend souvent des personnes quelque peu ignorantes de l'informatique demander devant un ordinateur: "Peut-il aussi faire du traitement de texte?" ou d'autres choses semblables. Ces personnes ne comprennent pas encore la différence qui existe entre un programme et un système d'exploitation et c'est pourquoi nous allons maintenant vous expliquer cette différence s'il vous est déjà arrivé de poser des questions semblables.

Nous allons prendre un exemple tiré de la vie courante, la tartine. Une tartine se compose d'une tranche de pain, de beurre et d'une garniture la plus appétissante possible. La tartine est un bon point de départ pour illustrer la différence entre système d'exploitation et programme.

Ce qui est indispensable pour faire n'importe quelle tartine, c'est la tranche de pain. Dans un système informatique, c'est

l'ordinateur qui constitue la tranche de pain. Il faut également obligatoirement du beurre sur votre tartine. Dans un système informatique: si vous possédez un ordinateur et que vous vouliez en faire quelque chose, il vous faut un système d'exploitation. Ce que vous placez ensuite sur votre tartine est votre problème et est fonction de vos goûts ou de votre faim. Il en va de même dans un système informatique. Les programmes que vous utilisez pour jouer sur l'écran ou pour faire du traitement de texte dépendent entièrement de vos besoins ou de vos goûts personnels.

Vous devriez maintenant comprendre que la question posée plus haut était mal formulée. En effet si un ordinateur possède un système d'exploitation, n'importe quel programme fait pour ce type d'ordinateur et sous ce système d'exploitation pourra tourner sur cet ordinateur.

II.1 Qu'est-ce qu'un programme?

Un programme est une suite d'instructions écrites les unes après les autres qui sont traitées dans l'ordre par l'ordinateur, l'une après l'autre. L'ordinateur a besoin que ces instructions lui soient fournies dans un langage qu'il puisse comprendre et traiter. Comme il est lui-même une machine, ce langage s'appelle le langage machine. Les programmes qui sont écrits en langage machine sont incompréhensibles pour des êtres humains normalement constitués. Mais l'ordinateur est content car ce langage lui est directement compréhensible et c'est également pourquoi il peut traiter particulièrement rapidement les suites d'instructions écrites dans ce langage.

Pour les êtres humains "normaux" existent encore des langages évolués ou langages de programmation. Vous connaissez certainement quelques-uns de ces langages. Les plus connus sont BASIC ou aussi FORTRAN ou PASCAL ou également un langage tout nouveau, MODULA 2. Avec ces langages de programmation évolués, ce sont des mots compréhensibles par des hommes qui sont utilisés comme instructions, traduits à l'intérieur de l'ordinateur en langage

machine puis traités par l'ordinateur.

Un programme n'est donc rien d'autre qu'une suite d'instructions qui indiquent exactement à l'ordinateur, étape par étape, ce qu'il doit faire. La longueur d'un tel programme peut varier de deux lignes d'instructions à un nombre presque infini de lignes d'instructions. Cela ne dépend que de la capacité mémoire de la mémoire centrale de l'ordinateur et des lecteurs de disquette.

Il ne suffit cependant pas qu'un programme réponde aux exigences de l'utilisateur, par exemple qu'il rende possible le traitement de texte. Il doit également prendre en charge à l'intérieur de l'ordinateur toute une série de tâches secondaires pour que les choses marchent correctement. C'est ainsi qu'il doit examiner sur quelle touche du clavier on vient juste d'appuyer, quel caractère doit être représenté sur l'écran ou sur l'imprimante et si et où les données ou programmes dont on a maintenant besoin figurent dans la mémoire de masse ou s'il faut charger des données de la disquette dans la mémoire de travail de l'ordinateur.

Il faut également s'assurer que les données chargées de la disquette arrivent bien dans l'endroit qui convient dans la mémoire de travail et qu'il reste suffisamment de place sur la disquette pour y écrire de nouvelles informations. Il faut encore tenir à jour un catalogue de la disquette, etc., etc.

II.2 Les sous-programmes

Comme vous le voyez, cela représente toute une masse de tâches secondaires. Et ces tâches secondaires doivent être à nouveau reprogrammées chaque fois pour tout programme. Pour chaque partie d'un programme doit être très nettement défini ce qui doit se passer dans l'ordinateur. Les choses sont encore compliquées par le fait que ces tâches doivent être résolues différemment pour les différents types d'ordinateur, aussi semblables qu'ils puissent paraître. Chaque ordinateur est en fait différent des autres. Si l'on écrivait donc d'une pièce un programme long, il devrait être entièrement réécrit pour chaque type d'ordinateur pour pouvoir fonctionner correctement.

On peut faire l'éconmie de ce travail inutile si l'on divise le programme en un programme principal et plusieurs sous-programmes. Les sous-programmes ne seront alors chargés dans la mémoire de travail de l'ordinateur que s'ils doivent effectivement être utilisés.

Pour adapter alors le programme à d'autres types d'ordinateur, il suffit de réécrire le ou les sous-programmes qui ne peuvent fonctionner sous cette forme sur le second ordinateur et la totalité du programme pourra tourner sans restriction sur le nouvel ordinateur. Un autre intérêt de cette technique des sous-programmes est que le programme principal ne sait plus du tout comment est résolue telle ou telle tâche parcellaire. Il se contente de donner une instruction, par exemple "sortir un texte" et le sous-programme résout ce travail. Que le texte soit sorti sur l'écran, sur une imprimante ou sur un télex n'a aucune importance pour le programme principal. Il donne simplement l'instruction et le travail est effectué.

Cette méthode fonctionne lorsque la transmission des données aux sous-programmes est standardisée dans le programme principal. On peut se représenter cela comme une course de relai où deux coureurs doivent toujours se transmettre le relai à un endroit très précisément fixé à l'avance. En langage informatique, un tel lieu

de transmission normalisé et déterminé précisément est appelé interface. Cette expression signifie qu'on peut à cet endroit retirer un sous-programme et le remplacer par un autre, sans que la fonction du programme principal n'ait à en souffrir. Tous les éléments restent adaptés les uns aux autres et fonctionnent sans problème.

Mais il est tout de même difficilement supportable pour chaque programmeur de devoir chaque fois réintégrer dans son programme les opérations internes. Cela représente beaucoup de travail et cela coûte beaucoup de temps et d'argent. C'est ce qu'on pensé également les hommes de DIGITAL RESEARCH qui ont donc eu l'idée d'écrire un programme standard pour les microordinateurs. Ils réunirent les programmes les plus courants qui sont nécessaires pour la gestion interne de l'ordinateur, ils définirent les interfaces et le mode de transmission des données entre programme principal et programme d'exploitation et ils réalisèrent ainsi un système d'exploitation pour les types d'ordinateur les plus différents.

Ce système d'exploitation ne peut bien sûr être utilisé que sur des ordinateurs disposant d'unités centrales semblables ou identiques car il faut bien que les instructions machine du programme puissent être comprises et traitées. Le premier système d'exploitation qui ait fourni de façon standard les nombreuses opérations de travail interne s'appelle CP/M. Cette abréviation signifie "Control Program for Micro-processors", soit programme de contrôle pour microprocesseurs. Ce système d'exploitation travaille avec les microprocesseurs du type 8080, 8085 ou Z80.

II.3 Les tâches de CP/M

CP/M résout pour l'essentiel les tâches de base suivantes: entrée de caractères, sortie de caractères, gestion de la place mémoire disponible sur la mémoire de masse, lecture des informations sur disquette et écriture de nouvelles informations sur disquette ou dans la mémoire de masse. Ces routines de service peuvent être utilisées par tous les programmes utilisateur d'une façon unique et

standardisée.

Pour que cela marche, CP/M doit cependant exécuter deux types différents de travail et c'est pourquoi il est divisé en deux grands blocs: la première partie, BDOS (Basic Disk Operating System = système d'exploitation de base pour disquettes) s'occupe de toutes les tâches qui peuvent toujours être résolues de la même façon, indépendamment du type de système informatique utilisé. La seconde partie du système d'exploitation, appelée BIOS (Basic I/O-System = système de base d'entrée/sortie) contient toutes les parties de programme qui sont adaptées et nécessaires pour un système informatique déterminé.

Chaque modèle d'ordinateur a en effet ses propres conceptions et méthodes pour résoudre certaines tâches déterminées et on ne peut pas prendre simplement le système d'exploitation d'une machine pour le faire tourner sur une autre. Avant qu'un système d'exploitation ainsi dérivé ne puisse fonctionner, le BIOS doit être adapté à la nouvelle machine. Vous ne devriez normalement avoir à cet égard aucun problème avec votre ordinateur car chaque ordinateur est livré (peut-on espérer) avec un système d'exploitation adapté.

II.4 CP/M et les différentes versions

Tout programme qui parvient à se maintenir un certain temps sur le marché de l'informatique connaît des améliorations à la suite desquelles il se présente sur le marché dans une nouvelle version. Il n'y a pas en effet de programmes sans erreurs et un certain nombre de ces erreurs ne sont découvertes que lorsqu'un programme est employé par de nombreux utilisateurs et que toutes les possibilités qu'il offre sont véritablement essayées.
Si un nombre suffisamment important d'indications d'erreurs parviennent au fabricant du programme, il corrige les erreurs et se résout tôt ou tard à sortir sur le marché une nouvelle version du programme. Dans sa version 2.2, qui est en fait le système d'exploitation standard pour tous les ordinateurs 8 bits, CP/M peut être considéré comme ne recélant pratiquement aucune erreur. La

nouvelle version CP/M 3.0 ne constitue donc pas une nouvelle édition pour éliminer des erreurs de la version précédente mais elle offre une sensible augmentation des possibilités et a reçu de nouvelles instructions supplémentaires. CP/M dut d'autre part être adapté aux ordinateurs et aux techniques informatiques qui devenaient toujours plus puissants.

II.5 L'interrogation CP/M

Je crois que cela suffit à présent. Nous allons pouvoir quitter un peu la théorie pour nous tourner vers la vie réelle, c'est-à-dire vers l'ordinateur et son système d'exploitation. Ce qu'il vous faut maintenant, c'est un ordinateur puissant, un lecteur de disquette et une disquette portant le système d'exploitation CP/M. Si vous avez tout cela, allumez votre ordinateur, introduisez la disquette avec le système d'exploitation CP/M dans le lecteur "A" ou lecteur "1" et verrouillez-le. Notez qu'il est vivement déconseillé d'allumer l'ordinateur lorsqu'une disquette figure dans le lecteur de disquette car l'impulsion électrique qui se produit alors peut entraîner des mouvements indésirables de la tête de lecture/écriture qui peuvent gravement endommager la disquette qui se trouvait dans le lecteur.

Bien, vous avez donc allumé maintenant votre ordinateur, vous avez placé votre disquette système CP/M dans le lecteur et vous avez entrepris la procédure de démarrage conformément à la description de votre manuel d'utilisation. Sur l'AMSTRAD CPC, vous devez allumer (ou éteindre puis allumer) la machine puis entre l'instruction:

ICPM

Vous obtenez le trait vertical par lequel commence l'instruction en appuyant sur les touches SHIFT + @.

Le lecteur de disquette commence à faire du bruit, la lampe de fonction s'allume et peu après vous voyez un message apparaître à

l'écran. Ce message appelé message initial se présente différemment sur chaque ordinateur. Cela vient du fait que ce message est sorti par la partie BIOS du système d'exploitation, c'est-à-dire par cette partie qui peut être spécialement adaptée à chaque ordinateur. Chaque fabricant d'ordinateur peut ainsi adapter cette partie à sa machine et disposer le message initial d'après ses propres conceptions. Un message initial CP/M se présente habituellement ainsi:

CPM 2.2 - AMSTRAD Consumer Electronics plc.
A>

Sur le CPC 6128, ce message se présente encore légèrement différemment car il s'agit là du message initial de CP/M 3.0.

CP/M Plus - AMSTRAD Consumer Electronics plc.
V 1.0, 61K TPA, 1 (2) Disc drive

Le message initial signifie pour vous que le système d'exploitation CP/M a été correctement chargé dans la mémoire de travail de l'ordinateur. Tout de suite après, CP/M vous envoie son message indiquant qu'il est prêt, qu'on appelle l'interrogation du système d'exploitation. Cette interrogation se présente ainsi:

A>

Le "A" indique à l'utilisateur qu'il travaille avec le lecteur "A" ou le lecteur "1" de son système. Le caractère ">" est le véritable message "prêt" de CP/M. Sur la ligne derrière le message "prêt", CP/M attend l'entrée d'une instruction. Ne faisons donc pas attendre CP/M trop longtemps et écrivons:

A>ABCDEFGH

Le curseur se trouve maintenant derrière la dernière lettre et rien ne se passe, la mer reste calme. La raison: CP/M ne sait pas si nous en avons déjà fini avec l'entrée de l'instruction ou si nous voulons encore écrire quelque chose à la suite. Pour mettre fin à

l'entrée, nous devons encore indiquer à CP/M que nous avons terminé notre entrée.

L'entrée d'une ligne doit être terminée sur l'ordinateur comme sur la machine à écrire par un retour de chariot. Cette touche s'appelle le plus souvent sur les ordinateurs, RETURN ou ENTER. Dès que vous appuyez sur cette touche, l'ordinateur sait que vous avez terminé l'entrée de l'instruction et il commence à exécuter l'instruction entrée. Appuyez maintenant sur la touche de retour de chariot, RETURN ou ENTER.

A>ABCDEFGH
ABCDEFGH?
A>

Bon, que s'est-il donc passé? CP/M a lu la ligne, n'a pas compris l'instruction et nous le dit. Le système d'exploitation nous indique qu'il ne comprend pas une instruction en répétant l'entrée et en la dotant d'un point d'interrogation. Le point d'interrogation signifie à peu près: "Qu'est-ce que cela signifie? Je ne peux rien en tirer."

Essayons maintenant de voir si l'erreur ne venait pas par hasard des majuscules. Entrez la même "instruction" que ci-dessus, mais en minuscules:

A>abcdefgh

Vous obtenez comme réponse:

A>abcdefgh
ABCDEFGH?
A>

Le système d'exploitation n'a donc pas compris non plus l'instruction en minuscules mais il a fait autre chose que la première fois. Avez-vous remarqué que la ligne d'instruction a été répétée en majuscules bien qu'elle ait été entrée en minuscules? C'est là une des particularités de CP/M que nous devrons d'ailleurs

prendre en compte dans un certain nombre de cas. CP/M convertit d'abord toutes les lettres que vous entrez en majuscules avant de les traiter.

Pour que vous voyiez que CP/M peut réellement travailler, nous allons maintenant entrer une instruction qui produira enfin un effet. Vous avez lu plus haut qu'une des tâches du système d'exploitation consiste à gérer les entrées sur la disquette dans un catalogue. CP/M organise à cet effet sur chaque disquette un catalogue (en anglais directory) qu'il affiche lorsqu'on le demande.

Si nous voulons examiner ce catalogue, nous devons indiquer au système d'exploitation que nous voulons le voir et qu'il doit présenter ce catalogue sur l'écran. L'instruction se compose des trois lettres DIR qui sont l'abréviation de directory. Entrez maintenant cette instruction, puis ENTER:

A>DIR

Vous voyez le catalogue de la disquette système d'exploitation sur l'écran. Examinez-le maintenant de plus près. Tout à gauche est indiqué le lecteur de disquette, "A" en l'occurence. A côté figurent les noms de fichiers suivis, après un espace de séparation, du type de fichier. Vous voyez très souvent le suffixe COM, par exemple DDT.COM, PIP.COM, ASM.COM. Ce type de fichier donne des indications sur le contenu du fichier. Pour le moment, ce sont les fichiers COM qui nous intéressent particulièrement car ils contiennent des programmes immédiatement exécutables.

COM est l'abréviation du mot anglais "Command" qui signifie instruction. Les fichiers ainsi désignés contiennent des instructions qui peuvent être immédiatement exécutées. Si vous tapez le nom d'un tel programme, sans le suffixe COM, à la suite de l'interrogation du système d'exploitation, le programme est directement chargé dans la mémoire de travail de l'ordinateur et exécuté. Il suffit donc d'entrer simplement PIP, puis ENTER et le programme tourne déjà.

II.6 On n'est jamais trop prudent

Si vous travaillez toujours actuellement avec votre disquette originale, nous devons vite nous préoccuper d'effectuer une copie de sécurité de cette disquette. Vous devriez prendre l'habitude de ne jamais travailler avec l'original d'un programme mais de toujours faire une copie du programme et de conserver soigneusement à l'abri l'original. Comme il convient pour un système d'exploitation qui est censé faciliter le travail de l'utilisateur, CP/M vous permet de réaliser des copies de sécurité.

II.7 Ce que vous savez déjà

* Vous savez en termes simples quelles tâches un système d'exploitation doit accomplir

* Vous savez ce qu'est un programme et à quoi servent les sous-programmes

* Vous connaissez les tâches de CP/M

* Vous savez ce qu'est l'interrogation CP/M et comment vous pouvez faire afficher le catalogue de votre disquette

* Vous avez pris la ferme résolution de réaliser de chaque disquette importante au moins une copie de sécurité

III. Le travail sous CP/M

III.1 La disquette système

Nous vous avons déjà indiqué que vous devez absolument effectuer des copies de vos disquettes originales les plus importantes. Nous allons maintenant vous montrer comment vous pouvez copier une disquette complète. Nous utiliserons à cet effet des instructions CP/M que nous n'expliquerons que plus tard. Vous devez utiliser deux programmes de la disquette CP/M pour pouvoir copier une disquette complète. Si vous entrez encore une fois DIR et que vous examiniez encore une fois attentivement le catalogue, vous trouverez entre autres deux programmes qui s'appellent:

SYSGEN.COM et PIP.COM pour CP/M 2.2 et

COPYSYS.COM et PIP.COM pour la version 3.0

Les choses se présentent toutefois différemment pour la firme AMSTRAD en ce qui concerne CP/M 3.0. Le fichier COM COPYSYS ne se trouve sur aucune des trois faces de disquette. Le programme pour copier le système, COPYSYS, est intégré dans le paquet de programmes DISCKIT3. Ce paquet de programmes est certainement très pratique mais qu'on nous permette la remarque suivante: cela n'aurait certainement pas coûté très cher de fournir malgré tout le fichier COPYSYS.COM. Il manque également le fichier COM pour le formatage d'une disquette. Je ne décris dans ce qui suit que la procédure normale sous CP/M 3.0, c'est-à-dire comme si les deux fichiers COM se trouvaient sur la disquette système. Reportez-vous au Chapitre 7 sous FORMAT et COPYSYS si vous voulez savoir comment utiliser ces instructions sur l'AMSTRAD.

Ces deux programmes vous permettent de faire différentes choses. Le programme SYSGEN (COPYSYS) vous permet de copier le système CP/M sur les deux premières pistes de chaque disquette. C'est pourquoi ces deux pistes sont également appelées les pistes système. Ce sont les premières à être lues lors de la mise sous tension de votre

ordinateur et le programme y figurant est chargé dans la mémoire de travail de votre ordinateur. Vous n'avez en tant qu'utilisateur d'ordinateur aucun accès direct aux pistes systèmes et c'est pourquoi vous ne pouvez trouver aucune CP/M.COM ou quoi que ce soit de semblable ni sur votre disquette ni dans son catalogue. Vous devez simplement savoir que les pistes système sont toujours réservées pour CP/M et qu'à cet endroit figurent toutes les parties de programme qui sont nécessaires à l'exploitation d'un ordinateur.

Mais si vous voulez maintenant copier une disquette entière, c'est-à-dire tous les programmes et également le système d'exploitation CP/M vous avez besoin d'une possibilité d'accéder à ces deux pistes système. Pour cela vous serez aidé par le programme SYSGEN (CP/M 2.2) ou en CP/M 3.0 par le programme COPYSYS. Ces deux programmes sont des programmes spéciaux qui vous permettent de recopier le contenu des pistes système d'une disquette sur une autre.

Avant que vous ne vous lanciez maintenant dans le travail de copie, vous devez vous préoccuper de trouver une disquette vide que vous veniez juste de formater (une disquette est formatée avec l'instruction CP/M FORMAT. Vous devez pour cela entrer cette instruction à la suite de l'interrogation CP/M et suivre les indications qui vous sont ensuite données à l'écran).

III.2 Copie avec un seul lecteur de disquette

Si tout est prêt, placez votre disquette originale CP/M dans le lecteur de disquette A et fermez la porte du lecteur de disquette. Si vous ne travaillez qu'avec un seul lecteur de disquette, vous aurez un peu plus de travail que si vous pouviez travailler avec deux lecteurs de disquette. Mais cela ne pose cependant pas de problème, si vous suivez les étapes suivantes. Pour copier votre système d'exploitation, introduisez la disquette originale dans le lecteur de disquette et entrez:

A>SYSGEN (ENTER)

Vous voyez alors apparaître à l'écran le message suivant:

Please insert SOURCE disc into drive A
then press any key:

où SOURCE se rapporte à votre disquette originale ou à la disquette sur laquelle vous voulez prendre le système d'exploitation. Une fois que la disquette se trouve dans le lecteur de disquette, appuyez sur ENTER et ce message apparaîtra:

Please insert DESTINATION disc into drive A
then press any key:

Entrez alors la nouvelle disquette dans le lecteur de disquette qui reprendra son travail après que vous ayez à nouveau appuyé sur ENTER. Peu de temps après vous verrez:

Do you wish to reconfigure another disc (Y/N)?

Si vous répondez ici Y pour oui, vous pouvez doter d'autres disquettes du système d'exploitation. Si vous répondez N pour non, vous êtes ramené à l'interrogation système.

III.3 Copie avec deux lecteurs et CP/M 3.0

Si vous travaillez avec deux lecteurs de disquette, la procédure fonctionne de la même façon si ce n'est bien sûr que la copie se fait d'un lecteur de disquette sur l'autre. Vous devez placer la disquette vide dans le lecteur de disquette B dont vous devez également refermer la porte. Lancez le système d'exploitation, entrez COPYSYS (SYSGEN pour CP/M 2.2) à la suite du message "prêt" de CP/M et appuyez sur la touche ENTER.

Les deux programmes se présentent ensuite avec leurs noms et leurs numéros de version et vous demandent d'entrer le lecteur de disquette à partir duquel doit se faire la copie. Comme votre disquette système d'exploitation CP/M se trouve dans le lecteur de

disquette A, entrez un "A" et on vous demande tout de suite si c'est bien la bonne disquette qui figure dans le lecteur de disquette A. Si oui, appuyez sur la touche ENTER et les questions-réponses continuent.

Vous devez ensuite indiquer le lecteur de disquette dans lequel se trouve votre disquette vierge, c'est-à-dire celle qui vient juste d'être formatée. Vous n'avez pas ici non plus besoin d'appuyer sur la touche ENTER car le programme veut tout de suite savoir si la disquette est vraiment dans le lecteur de disquette et vous demande explicitement en confirmation d'appuyer sur la touche ENTER. Une fois que vous avez suivi toutes les instructions et que vous avez entré votre dernier ENTER, le lecteur de disquette B commence à tourner et le système d'exploitation CP/M est écrit sur la nouvelle disquette. Vient ensuite la question de savoir si CPM3.SYS doit également être copié. Il s'agit de programmes utilitaires importants et vous devriez les avoir sur toute copie de votre disquette système. Dès que le lecteur de disquette B s'est à nouveau arrêté, le travail est terminé et vous pouvez introduire une autre disquette pour y copier le système d'exploitation. Comme nous avons cependant encore l'intention d'utiliser notre disquette de travail, appuyez sur la touche ENTER pour mettre fin au programme de copie système.

III.4 Examiner le catalogue

Vous avez copié quelque chose de A dans B et vous êtes curieux de voir ce qui se trouve sur la disquette B. Pour satisfaire votre curiosité, entrez directement à la suite de l'interrogation CP/M:

DIR B:

Si vous n'avez pas oublié l'espace après DIR, vous verrez que le lecteur de disquette B commence à tourner et que vous recevez un message à l'écran. Avec DIR B:, vous avez demandé à CP/M de vous montrer le contenu de la disquette figurant dans le lecteur de disquette B. Mais comme vous n'avez jusqu'ici copié que le système

d'exploitation mais encore aucun fichier, vous voyez apparaître le message NO FILE. FILE signifie fichier et NO FILE signifie donc qu'il n'y a rien sur votre disquette. Comme vous vous le rappelez certainement, les programmes des deux premières pistes système de chaque disquette sont dissimulés. Vous n'y avez donc aucun accès direct et vous ne pouvez pas non plus les voir dans le catalogue.

III.5 Copie avec PIP et deux lecteurs de disquette

Mais comme nous voulons copier entièrement la disquette système CP/M et pas seulement le système d'exploitation, nous allons maintenant utiliser un autre programme. Ce second programme a un nom très long en français: processeur pour les échanges de données entre unités périphériques, en anglais Peripheral Interchange Processor, ce qui donne l'abréviation PIP.

Vos deux disquettes se trouvent toujours dans les lecteurs de disquette et vous tapez maintenant PIP puis vous appuyez sur la touche ENTER. L'ordinateur charge maintenant PIP dans la mémoire de travail et le programme se présente alors avec son propre message d'interrogation:

```
A>PIP
*
```

L'étoile vous indique que PIP est prêt à recevoir vos instructions. Ce programme est vraiment très utile et puissant. Vous découvrirez plus précisément les nombreuses possibilités de PIP au Chapitre 6.

Avant que nous n'entrions maintenant une instruction pour PIP, demandons-nous d'abord ce que nous attendons de lui. Les fichiers de la disquette figurant dans le lecteur de disquette A doivent être transférés sur la disquette figurant dans le lecteur de disquette B. On pourrait également dire que "B:" recevra tous les fichiers qui sont stockés sur "A:".

Vous devez lire attentivement cette dernière phrase et essayer de

bien la retenir. Elle est décisive pour le travail avec PIP. Une fois que tous les fichiers ont été transférés de A sur B, le contenu des deux disquettes est identique. C'est pourquoi on utilise le signe égale dans les instructions PIP. Le seul problème qui demeure est de savoir comment nous allons dire à PIP qu'il doit prendre tous les fichiers et non pas seulement certains d'entre eux.

Ce cas est prévu. L'étoile figurant à la place d'un nom de fichier signifie en effet "tous les fichiers". Vous pouvez comparer l'étoile à un Joker pouvant remplacer n'importe quelle carte dans un jeu de cartes. Mais comme les fichiers ne se distinguent pas seulement les uns des autres par leur nom de fichier mais également par le type de fichier, c'est-à-dire par les trois lettres placées après le point, vous devez entrer, pour copier tous les fichiers, étoile/point/étoile. Sur l'écran, cela se présente ainsi:

```
B:=A:*.*
```

Comme nous voulons être sûr que tous les fichiers soient vraiment transférés correctement, nous ferons immédiatement vérifier chaque fichier pour voir s'il a été transféré correctement. Nous pouvons confier ce travail à PIP en ajoutant à la suite de notre instruction un "V" entre crochets. "V" est mis pour "Verify". Mais pour que PIP comprenne bien cette instruction, elle doit figurer entre crochets.
Une fois que vous avez entré l'instruction complète,

```
A>PIP
*B:=A:*.COM
```

vous voyez à l'écran le catalogue CP/M 3.0 suivant:

```
A: CCP      COM : DATE     COM : DEVICE   COM : DIR      COM : DUMP     COM
A: ED COM : ERASE COM : GENCOM COM : GENCPM COM : GET COM
A: LIB COM : LINK COM : MAC COM : PATCH COM : INITDIR COM
A: PUT COM : RENAME COM : RMAC COM : SAVE COM : SET COM
A: HEXCOM COM : SETDEF COM : SHOW COM : SID COM : SUBMIT COM
A: TYPE COM : XREF COM : COPYSYS COM : FORMAT COM : HELP COM
A: PIP COM
```

Comme vous le voyez pendant le déroulement de la procédure de copie, PIP vous indique au fur et à mesure quel fichier il est en train de copier. Une fois le travail terminé, PIP se présente à nouveau avec son caractère "prêt", l'étoile. Si vous n'avez pas d'autre travail pour PIP, et je crois que pour le moment nous n'avons plus rien à demander à PIP, appuyez sur la touche ENTER et le système d'exploitation se présente à nouveau avec son interrogation A.

Si vous n'avez pas une totale confiance dans la technique, vous pouvez contrôler avec DIR B: si tous les fichiers sont véritablement arrivés sur le lecteur de disquette B. Si tout est satisfaisant, retirez la disquette du lecteur de disquette A, c'est-à-dire votre disquette originale, replacez-la dans sa boîte et placez cette disquette dans un endroit où elle ne risquera rien. Vous ne travaillerez à l'avenir qu'avec la copie que vous venez de réaliser, ce qui vous évitera bien des problèmes s'il arrive quoi que ce soit à la disquette système lors de manipulations. En effet: une disquette pour effectuer une copie ne vous coûte que quelques francs (les disquettes 3 pouces pour l'AMSTRAD sont un peu plus chères) alors qu'un nouveau système d'exploitation CP/M vous coûterait plusieurs centaines de francs.

III.6 Règles pour les noms de fichier

Nous avons jusqu'ici parlé à plusieurs reprises des fichiers, nous avons également consulté les noms de fichier dans le catalogue sur

l'écran mais nous ne savons toujours pas ce qu'est un fichier ni à quoi cela correspond.

Comme vous le savez, sous CP/M, chaque fichier a un nom. Ce nom est déjà important pour vous, de façon à ce que vous sachiez ce qu'il y a dans un fichier mais il est également important pour CP/M pour qu'il puisse retrouver ce fichier lorsque vous l'appelez. Malheureusement les constructeurs de CP/M ont introduit une limitation qui personnellement ne me convient absolument pas.

Un nom de fichier sous CP/M doit se composer au maximum de huit caractères, d'un point et de trois caractères pour ce qui est appelé la marque. Cela signifie que vous devez trouver pour vos fichiers des noms qui comportent au maximum huit caractères et qui aient malgré tout un sens suffisamment évocateur pour vous permettre de deviner le contenu du fichier correspondant.
Les caractères autorisés pour un nom de fichier sont les 26 lettres de l'alphabet ainsi que les signes plus, moins, trait diagonal (slash), pourcentage et dollar en n'importe quelle position du premier nom ou de la marque. Les signes "inférieur à", "supérieur à", point, virgule, double point, égale, trait d'union, étoile, point d'interrogation, crochet ouvert, crochet fermé et point d'exclamation ne doivent pas être utilisés dans le nom de fichier ni dans la marque car ces signes ont une signification précise pour CP/M, de sorte que leur utilisation pourrait se traduire par des effets indésirables.

Mon expérience de longues années de travail avec CP/M et d'autres programmes ne peut que me conduire à vous conseiller vivement de toujours donner à vos fichiers un nom parlant qui vous permette de vous souvenir de leur contenu. Surtout si vous avez de nombreux programmes sur votre disquette, vous ne pourrez rien tirer d'un nom tel que XK2512AC.ABC. Vous n'avez par contre pas de précautions particulières à prendre dans le choix du nom de fichier car il ne peut pas y avoir simultanément sur une même disquette deux fichiers ayant le même nom et la même marque. Toutefois il peut y avoir sur une disquette des fichiers ayant le même nom mais des marques différentes.

III.7 Marque de fichier

Vous pouvez organiser la marque d'un nom de fichier comme vous le voulez, c'est-à-dire que vous pouvez appeler chaque fichier comme vous le voulez mais vous devriez tenir compte des abréviations-type de CP/M. Vous trouverez à la fin du présent chapitre une table des abréviations usuelles sous CP/M et de leur signification. Vous y trouverez par exemple des programmes avec la marque COM. C'est l'abréviation de COMMAND et cela signifie que ces programmes contiennent du code d'instruction directement exécutable.
Vous vous rappelez que PIP et SYSGEN ou COPYSYS font partie de ces programmes COM. Vous devriez d'autre part vous garder de donner à des fichiers la marque BAK ou $$$. BAK signifie "back up" ce qui veut dire copie de sécurité. Cette marque est utilisée par l'éditeur CP/M pour désigner les copies de sécurité de fichiers. Un fichier avec les trois signes dollar est un fichier provisoire qui est créé par un programme et qui sera supprimé à la fin du déroulement du programme. PIP crée par exemple de tels fichiers provisoires et les supprime totalement à la fin de son travail. Il vaut donc mieux faire attention.

Il est également déconseillé d'utiliser des noms de fichier identiques et de n'utiliser que la marque pour compter les différents fichiers. Exemple: vous avez un texte et vous appelez le premier fichier TEXTE.1, le second TEXTE.2 et le troisième fichier TEXTE.3. Ce qui peut vous sembler parfaitement logique sur le moment vous attirera des ennuis par la suite. En effet, dès que vous aurez traité le premier fichier, votre fichier original deviendra TEXTE.BAK et c'est la version que vous venez de modifier qui s'appellera maintenant TEXTE.1. Si vous travaillez maintenant sur le fichier TEXTE.2, ici aussi le fichier original deviendra TEXTE.BAK et la version améliorée TEXTE.2.

Comme il ne peut y avoir sous CP/M deux fichiers sur une même disquette portant exactement le même nom et la même marque, le système d'exploitation aura cependant cette fois-ci d'abord supprimé votre premier fichier BAK et l'aura remplacé par le second fichier BAK. Vous n'avez donc plus maintenant de possibilité de

retrouver le texte original de votre premier fichier TEXTE.1.

Pour contourner ce problème, il vaudrait mieux appeler par exemple votre premier fichier TEXTE1.TXT, votre second TEXTE2.TXT et le troisième TEXTE3.TXT.

Si vous modifiez par la suite vos fichiers, vous aurez pour chaque fichier un fichier de back up différent et vous aurez ainsi la possibilité de retrouver à tout moment le texte de la version précédente.

III.8 Retrouver un fichier

Si vous utilisez votre ordinateur de façon intensive et que vous créiez de nombreux fichiers différents, vous rencontrerez peut-être bientôt un nouveau problème. Vous risquez en effet de ne plus retrouver aussi facilement dans votre catalogue les fichiers que vous cherchez. Il pourrait donc être très intéressant pour vous de pouvoir examiner combien de fichiers portant le nom TEXTE.TXT figurent déjà sur la disquette.

Vous avez deux possibilités pour faire rechercher ces fichiers. Dans les deux versions du système d'exploitation, ce sont ici l'étoile et le point d'interrogation qui vous viendront en aide. Nous avons déjà rencontré l'étoile en tant que ce que nous avons appelé Joker lors de notre opération de copie avec PIP. Le point d'interrogation travaille de façon similaire. Alors que l'étoile remplace tous les caractères du nom de fichier ou de la marque, le point d'interrogation remplace un caractère quelconque dans une position déterminée.

L'entrée "Texte?" par exemple fera rechercher tous les fichiers dont le nom est "Texte" ou "Texte" plus un autre caractère. C'est ainsi que les trois fichiers "TEXTE1", "TEXTE2" et "TEXTE3" seraient ainsi trouvés. Si figurait en outre également un fichier "TEXTE" sur la même disquette, celui-ci serait également trouvé et affiché. Cela vient du fait que CP/M réserve toujours huit

caractères pour un nom de fichier. Si le nom de fichier que vous entrez est plus court, CP/M remplit les emplacements restants avec des caractères espace. Les espaces sont donc à cet égard des caractères tout aussi valables que les lettres ou les chiffres.

III.9 Recherche avec le point d'interrogation (?)

Comme le point d'interrogation peut être mis à la place de n'importe quel caractère, ce ne sont pas seulement les fichiers "TEXTE1", "TEXTE2" et "TEXTE3" qui seront trouvés, mais également le fichier "TEXTE". L'emploi du point d'interrogation ne se limite cependant pas seulement à un caractère. Vous pouvez tout aussi bien entrer jusqu'à huit points d'interrogation pour le nom de fichier et jusqu'à trois points d'interrogation pour la marque. Dans ce cas limite cependant, vous pouvez vous éviter de taper autant de caractères en entrant une étoile pour le nom de fichier et une étoile pour la marque. Pour le système d'exploitation, cela revient exactement au même que vous entriez huit points d'interrogation pour le nom de fichier ou que vous entriez une étoile. De façon interne, une étoile est de toute façon convertie en huit points d'interrogation avant que la recherche ne commence.

Vous pouvez également effectuer une recherche plus précise avec l'étoile en écrivant devant l'étoile une ou plusieurs des lettres initiales des fichiers recherchés. Si vous entrez toutefois une étoile suivie de quelques lettres, la recherche ne fonctionnera pas. Vous pouvez utiliser cette méthode d'entrée de noms de fichier avec point d'interrogation ou étoile en liaison avec les instructions DIR, TYPE, STAT, FILECOPY et PIP.

III.10 Ce que vous savez déjà

* Vous pouvez effectuer maintenant une copie d'une disquette système de façon à bien conserver votre original

* Vous savez employer l'étoile et le point d'interrogation pour remplacer des lettre manquantes, ce qui vous facilite le travail

* Vous savez qu'on doit toujours donner aux fichiers des noms parlants de façon à ce qu'on puisse les retrouver

IV. Instructions intégrées

Nous avons découvert dans le chapitre précédent quelques fonctions de base du système d'exploitation CP/M. Mais nous sommes encore loin du compte. Dans le présent chapitre nous nous occuperons plus précisément des instructions CP/M intégrées qui sont: USER, DIR, DIRSYS (CP/M 3.0), REN, RENAME (CP/M 3.0) et TYPE.

Nous nous intéresserons en outre plus tard à ce qu'on appelle les programmes transitoires (ce sont des programmes CP/M qui apparaissent dans le catalogue et qui sont stockés sous la forme de fichiers COM). Vous augmenterez en outre vos connaissances sur les instructions DIR et PIP que vous avez déjà rencontrées dans le chapitre précédent.

IV.1 Instructions, paramètres et options

Avant de vous expliquer d'autres instructions CP/M, il vous faut encore apprendre la différence fondamentale entre une instruction, un paramètre et une option. Une instruction est un mot d'instruction qui ordonne à CP/M d'exécuter une action déterminée. Un paramètre est en règle générale un nom de fichier qui indique à quel fichier se rapporte l'instruction. Une option figure sous CP/M entre crochets et elle peut être laissée de côté comme son nom le suggère. Mais si une option est entrée, elle modifie d'une façon déterminée la fonction de l'instruction qui vient d'être entrée.

Vous avez déjà rencontré un exemple d'option lorsque nous avons copié votre disquette système avec PIP. L'option était "V" et elle avait pour effet de faire que PIP vérifie tous les fichiers copiés.

Suivant l'instruction dont il s'agit, des paramètres sont entrés ou non avec l'instruction. Si un paramètre est nécessaire pour une quelconque instruction et si vous n'en avez entré aucun, CP/M vous demandera expressément d'entrer ce paramètre. Vous n'avez donc pas à craindre que quelque chose ne fonctionne pas correctement.

En entrant les instructions et les paramètres, vous devez faire attention à ce qu'il y ait bien toujours au moins un espace entre l'instruction et le paramètre. C'est d'ailleurs le seul endroit où un espace est autorisé. Vous ne devez en effet utiliser d'espaces ni dans l'instruction, ni dans le paramètre, ni dans l'option.

Vous n'entrerez normalement après une instruction qu'un ou deux noms de fichier comme paramètres. Vous n'êtes toutefois pas obligé de vous limiter à aussi peu d'indications. Si vous avez suffisamment d'idées pour cela ou si votre application le réclame, vous pouvez également écrire sur toute la ligne, commencer une nouvelle ligne avec Control E (^E) et continuer sur cette nouvelle ligne. CP/M lira ces deux lignes comme une ligne unique d'instructions.

IV.2 Les instructions intégrées

Dès que CP/M a été chargé de la disquette dans la mémoire de travail de l'ordinateur, vous disposez de deux sortes d'instructions. Les instructions intégrées se trouvent dans la mémoire de travail de votre ordinateur et elle peuvent y être appelées immédiatement et très rapidement pour accomplir certaines tâches. Les autres instructions se trouvent sous forme de fichiers sur votre disquette et vous pouvez en consulter la liste en examinant le catalogue de la disquette. Ces programmes peuvent être copiés ou supprimés ou même, si vous disposez des connaissances nécessaires, modifiés exactement comme des fichiers normaux.
Cette subdivision de CP/M a été établie parce que les deux pistes système de chaque disquette qui sont réservées pour CP/M sont loin d'offrir une place suffisante pour tous les programmes utilitaires. Pour votre travail avec l'ordinateur il est cependant indifférent que vous appeliez une instruction intégrée ou une instruction transitoire. Si vous réclamez par exemple avec l'instruction PIP des actions très particulières, CP/M chargera automatiquement le fichier PIP.COM dans la mémoire de travail et exécutera votre instruction.

Le seul inconvénient pour vous est un peu de perte de temps occasionnée par la lecture du fichier sur la disquette. Le système d'exploitation CP/M 2.2 possède en tout cinq instructions intégrées et la version 3.0 six instructions intégrées, quatre de ces instructions recevant une extension d'un fichier COM portant le même nom. Si vous appelez sous CP/M une de ces instructions, vous n'êtes pas obligé d'entrer chaque fois le nom complet de l'instruction. Vous pouvez abréger l'entrée de l'instruction comme indiqué dans la table ci-dessous:

Instruction	Abréviation	Fonction
DIR	DIR	affiche le catalogue
DIRSYS	DIRS	affiche la liste des fichiers système
ERASE	ERA	suppression de fichiers
RENAME	REN	modification du nom de fichier
TYPE	TYP	affiche fichier de texte
USER	USE	modifie zone utilisateur

Dans la seconde version vous êtes obligé d'utiliser les abréviations au lieu du nom in extenso des instructions car le système d'exploitation ne comprend que les abréviations.

Pour être exact nous devrions dire qu'il existe encore une instruction intégrée de CP/M qui n'a cependant pas de nom. Vous pouvez utiliser cette instruction pour commuter du lecteur de disquette A sur le lecteur de disquette B ou sur n'importe quel autre lecteur de disquette. Si au lieu du lecteur de disquette A vous préférez que le lecteur de disquette annoncé soit le lecteur de disquette B, tapez simplement:

B:(ENTER)

et CP/M fera du lecteur de disquette B le lecteur de disquette annoncé.

Si vous travaillez maintenant avec le lecteur de disquette B alors que votre disquette CP/M se trouve dans le lecteur de disquette A, vous pouvez malgré tout accéder aux programmes CP/M de la disquette A. Il vous suffit pour cela d'entrer le nom du lecteur de disquette suivi d'un double point avant l'instruction. Cela peut se présenter ainsi:

B>A:DIR

Vous pouvez également placer la lettre du lecteur de disquette devant un paramètre:

B>DIR A:FICHIER.XXX

IV.3 USER

Ce programme intégré vous permet de diviser un support de stockage en 16 zones numérotées de 0 à 15. Une telle division peut être très utile lorsque plusieurs utilisateurs se partagent un ordinateur et que leurs fichiers ne doivent pas être mélangés. Une autre possibilité est de conserver chaque fois dans une zone utilisateur différente les différents programmes avec les fichiers qui leur correspondent.

Vous pourriez ainsi organiser une zone pour vos fichiers de texte, une zone pour vos fichiers BASIC, une zone pour vos lettres commerciales, etc., etc. Cependant cette division d'un support de stockage n'acquiert un intérêt réel que lorsqu'on utilise un disque dur qui offre beaucoup plus de place qu'une disquette ce qui rend la gestion de tous les fichiers déjà nettement plus difficile.

La division en zones rend également possible de faire figurer sur le même support de stockage deux fichiers portant exactement le même nom et la même marque, à condition bien sûr que chaque fichier

figure dans une zone différente.

IV.3.1 Zones USER sur CP/M 2.2

La possibilité de différentes zones utilisateur, reprise dans la version CP/M 2.2, a été tirée du système d'exploitation multi-postes MP/M. Chaque zone travaille comme une disquette indépendante, c'est-à-dire que les fichiers sont créés séparément dans chaque zone utilisateur et les programmes nécessaires à leur exploitation doivent tous être disponibles dans cette zone utilisateur. La commutation d'une zone utilisateur à l'autre s'effectue avec l'instruction:

USER n

Le "n" est mis ici pour un nombre entre 1 et 15. Tout travail effectué à l'intérieur d'une zone reste sans effet sur les fichiers figurant dans les autres zones. Par exemple, l'instruction:

ERA *.*

supprime tous les fichiers, mais uniquement à l'intérieur d'une zone utilisateur. Si vous travaillez avec deux lecteurs de disquette et que vous copiez des fichiers d'un lecteur de disquette sur l'autre, ces fichiers apparaissent normalement sur l'autre lecteur de disquette dans la même zone utilisateur. Si vous copiez par exemple la zone 3A avec PIP sur B, les fichiers atterriront dans la zone 3B.

Après chaque redémarrage à froid de CP/M, le système d'exploitation commence dans la zone utilisateur zéro. Si vous voulez travailler dans une autre zone, vous devez d'abord entrer la nouvelle zone utilisateur. Malheureusement, l'interrogation CP/M ne vous indique

pas dans quelle zone utilisateur vous vous trouvez pour le moment. Vous ne pouvez l'apprendre qu'avec l'instruction STAT.

IV.3.2 Zones USER sous CP/M 3.0

La zone utilisateur zéro acquiert ici une signification particulière. Les programmes et les fichiers qui figurent dans cette zone et qui ont été déclarés comme fichiers système peuvent être appelés et utilisés à partir de n'importe quelle zone utilisateur. Nous vous donnerons plus de détails à ce sujet dans une section ultérieure. Dès que vous vous trouvez dans une zone utilisateur et que vous y créez un nouveau fichier, CP/M note automatiquement dans quelle zone utilisateur ce fichier a été créé.

Chaque fois que vous allumez votre ordinateur et que vous chargez CP/M, vous vous trouvez dans la zone utilisateur 0. Lorsque vous voulez travailler dans une autre zone utilisateur, vous pouvez entrer l'instruction USER. Par exemple pour travailler dans la zone utilisateur numéro 3, entrez:

USER 3

Faites attention à bien séparer USER du nombre correspondant par un espace. Dès que CP/M a commuté sur la nouvelle zone utilisateur, vous voyez que quelque chose se modifie à l'écran. L'interrogation CP/M ne se présente plus maintenant simplement comme un "A" mais vous voyez aussi devant le A l'indication de la zone utilisateur, en l'occurrence donc un "3". Vous voulez certainement maintenant travailler également avec le second lecteur de disquette et pouvoir y commuter entre les différentes zones utilisateur. Pour parvenir par exemple dans la zone utilisateur du lecteur de disquette B, entrez:

USER 3B:

et vous voyez alors l'interrogation CP/M qui se présente ainsi:

3B>

Si vous entrez maintenant l'instruction DIR pour consulter le catalogue de votre disquette, vous voyez "NO FILE". A condition bien sûr que vous n'ayez encore créé aucun fichier dans cette zone utilisateur. Vous pouvez accéder à toutes les instructions intégrées de CP/M à partir de n'importe quelle zone utilisateur. Pour revenir à la zone utilisateur 0, entrez:

USER 0

et vous atterrissez à nouveau dans la zone utilisateur la moins élevée. Les programmes qui y figurent et qui doivent être disponibles pour toutes les zones utilisateur peuvent être préparés pour cela avec l'instruction SET. Je vous montrerai dans une section ultérieure comment cela fonctionne. Vous pouvez cependant effectuer la commutation entre les zones utilisateur de façon encore plus simple en entrant votre demande de changement de la façon suivante:

B1:

En ce qui concerne les zones utilisateur, il faut encore prendre en compte le point suivant: lorsque vous copiez avec PIP un fichier de "A" à "B" ou inversement, ce fichier n'est copié que dans la même zone utilisateur.

IV.4 DIR

Vous connaissez déjà l'instruction DIR que vous avez utilisée au chapitre précédent pour examiner le catalogue de votre disquette. L'instruction DIR est identique pour les deux versions CP/M si ce n'est qu'elle se compose de deux programmes dans la version 3.0. Le premier programme est l'instruction intégrée DIR, qui est identique à celle de CP/M 2.2 et le second programme est DIR.COM qui fournit des fonctions considérablement étendues.

Nous nous intéresserons dans un premier temps à l'instruction DIR telle qu'elle peut être utilisée dans les deux versions.

Si vous entrez l'instruction DIR sans rien d'autre à la suite, tous les fichiers de votre disquette dans cette zone utilisateur précise apparaissent. Si donc vous entrez DIR alors que vous êtes dans la zone utilisateur 3, seuls les fichiers figurant dans la zone utilisateur 3 apparaîtront. Il arrive parfois sous CP/M 3.0 qu'il y ait tellement de fichiers dans une zone utilisateur qu'ils soient loin de tous rentrer dans une page d'écran.

Dans ce cas, la sortie sur l'écran du catalogue s'arrête jusqu'à ce que vous la relanciez en appuyant sur une touche quelconque. Si vous voulez arrêter la sortie du catalogue, appuyez simplement Control C (^C).

Si vous travaillez pour le moment avec le lecteur de disquette A mais que vous souhaitiez voir si un fichier déterminé figure dans le lecteur de disquette B, vous disposez de deux possibilités pour examiner le catalogue de B. Vous tapez:

B:
DIR

ou, ce qui va plus vite:

DIR B:

Avantage supplémentaire de la seconde manière: elle vous permet de rester sur le lecteur de disquette A. Si vous préférez sortir sur imprimante le catalogue de votre disquette, entrez:

DIR B:

puis Control P (^P) et ENTER. Le catalogue complet sera alors sorti sur imprimante. Une fois l'impression terminée, appuyez à nouveau Control P et l'imprimante interrompra son travail.

IV.4.1 DIR avec des paramètres

Il n'y a pas de quoi s'extasier des fonctions simples de DIR et c'est bien ce qu'on attend d'un système d'exploitation normal. DIR devient cependant très utile et puissant lorsqu'on commence à utiliser les nombreuses possibilités supplémentaires qu'il recèle. Supposons par exemple que votre catalogue compte une masse importante d'entrées et que vous souhaitiez rechercher un fichier déterminé. Au lieu de lire maintenant tous les fichiers et de rechercher ainsi où se trouve votre fichier, il vous suffit d'entrer:

DIR B:TEXTE.TXT

et ce nom sera affiché si le fichier figure effectivement sur le lecteur de disquette B. Si ce fichier n'y figure pas, vous recevrez de CP/M un message d'erreur.

Si vous voulez utiliser les fonctions étendues de DIR, vous devez

travailler, dans la version 3.0, avec le programme DIR.COM. Cela signifie que si vous travaillez sur le lecteur de disquette B mais que DIR.COM se trouve sur le lecteur de disquette A, vous ne devez pas oublier d'entrer également l'indication du lecteur de disquette avant DIR. Cela se présente par exemple ainsi:

A:DIR (options)

Vous vous souvenez certainement que l'étoile peut nous enlever beaucoup de travail de frappe lors d'une recherche. Avec l'instruction DIR combinée avec l'étoile, vous pouvez faire afficher un certain type de fichiers. Cela fonctionne ainsi:

DIR *.COM

Si vous entrez l'instruction sous cette forme, vous verrez à l'écran tous les fichiers qui possèdent un COM comme marque de fichier.

IV.4.2 DIR étendu sous CP/M 3.0

Vous pouvez dans la nouvelle version de CP/M faire rechercher de la même manière plusieurs types différents de fichiers. L'instruction correspondante se présente ainsi:

DIR *.COM *.SUB

Pour que cela fonctionne vraiment, vous devez expressément demander à CP/M l'utilisation de DIR.COM.

Vous obtenez cela soit en entrant la désignation du lecteur de disquette A: ou B: devant le DIR, soit en écrivant une option entre crochets à la suite du nom de fichier. Les deux options possibles

dans ce cas sont FULL ou DIR.

IV.4.3 DIR et ses options

Vous pouvez indiquer avec DIR un total de 18 options différentes. DIR a ainsi une puissance comparable à celle de PIP et il fait partie des programmes les plus utiles de CP/M. Vous n'entrerez normalement que rarement simultanément plus d'une ou deux options. Les options de DIR figurent toujours entre crochets.

Si plus d'une option est indiquée, les options doivent être séparées entre elles par des virgules ou des espaces. S'il n'y a pas de confusion possible, vous pouvez aussi abréger le nom d'une option à deux lettres. Vous pouvez également simplifier l'entrée de l'instruction en négligeant le crochet fermé si c'est le dernier caractère de la ligne d'instruction.

IV.4.4 DIRSYS

Lorsque vous examinez un catalogue sous CP/M 3.0, vous avez certainement déjà remarqué qu'au bord inférieur de l'écran apparaît un message qui se présente ainsi:

SYSTEM FILE (S) EXIST

CP/M vous indique ainsi que sur les disquettes figurent encore, outre les fichiers dont vous voyez la liste, ce qu'on appelle des fichiers système qui figurent dans la zone utilisateur 0 et qui sont accessibles et utilisables à partir de n'importe quelle zone utilisateur. Si vous voulez examiner vos fichiers système, entrez l'instruction DIRSYS ou en abrégé simplement DIRS. Sous la liste de ces fichiers vous recevrez alors le message indiquant que des fichiers "non-système" figurent également sur la disquette.

Bien entendu, vous disposez avec l'instruction DIRSYS des mêmes possibilités qu'avec l'instruction DIR, vous pouvez donc également utiliser l'étoile et le point d'interrogation, exactement comme nous l'avons expliqué plus haut.

IV.5 ERASE

La place disponible sur la disquette ainsi que le nombre d'entrées possible dans le catalogue de chaque disquette sont limités. Il arrivera tôt ou tard que vous n'aurez plus de place sur la disquette et que vous serez contraint de supprimer un ou plusieurs fichiers sur cette disquette. Le système d'exploitation CP/M permet de supprimer des fichiers avec l'instruction ERA.

Entrez l'instruction de la façon suivante:

ERASE D:NOM

où D est mis pour le nom du lecteur de disquette et NOM pour le nom de votre fichier. Si vous travaillez par exemple avec le lecteur de disquette A et que vous vouliez y supprimer un fichier, vous n'avez pas besoin d'entrer également le nom du lecteur de disquette.
Pour supprimer en une fois plusieurs fichiers, vous pouvez naturellement employer ici aussi les deux Jokers étoile et point d'interrogation. Il convient toutefois de bien réfléchir à ce que vous faites avant d'utiliser ces instructions puissantes. Il peut en effet arriver trop facilement que vous supprimiez des fichiers que vous auriez voulu garder et que vous vous en mordiez ensuite les doigts de rage.

Si vous voulez par exemple supprimer tous les fichiers qui possèdent un BAK comme marque, entrez:

ERA *.BAK

et, sous CP/M 2.2, tous les fichiers possédant la marque correspondante seront supprimés. Avec les fichiers backup, comme dans le cas de notre exemple, une erreur n'est certainement pas si dramatique. Mais imaginez un peu que vous ayez entré l'instruction suivante:

ERA *.*

et que vous ayez ensuite appuyé sur ENTER sans plus réfléchir. Eh bien toutes les données qui figuraient sur votre disquette auront disparu.

Mais comme cette instruction si puissante peut avoir de tels effets destructeurs, les inventeurs de CP/M ont introduit une étape de sécurité. En effet, dès que vous effectuez une suppression avec deux étoiles (*.*), on vous demande par précaution de confirmer que c'est vraiment votre intention:

ALL (Y/N)

(=tout (oui/non)). Si vous voulez vraiment tout supprimer, entrez un Y pour oui puis appuyez sur la touche ENTER. Dans le cas contraire, entrez un N pour non et vous aurez sauvé la situation.

IV.5.1 Suppression avec ERA sous CP/M 3.0

Le directeur de DIGITAL RESEARCH a certainement un jour supprimé lui-même par mégarde tous les fichiers contenant ses comptes secrets. En effet, dans la nouvelle version de CP/M, le programme vous demande déjà lorsque vous n'entrez qu'un Joker:

ERA *.BAS

de confirmer si tout doit bien être supprimé:

ERASE *.BAS (Y/N)?

Avant que vous n'entriez "Y" pour répondre oui, vous devriez vraiment réfléchir très sérieusement à ce que vous êtes en train de faire. Etes-vous absolument sûr que l'instruction ait été formulée correctement et que c'est exactement ce type de fichiers que vous voulez supprimer? Une fois que vous aurez tapé votre "Y", vous ne pourrez plus revenir en arrière. Les fichiers seront supprimés. Faites à cet égard surtout attention aux éventuelles fautes de frappe. On peut par exemple aisément écrire PAS au lieu de BAS. Au lieu de vos fichiers BASIC ce seraient alors vos fichiers PASCAL qui seraient par exemple supprimés...

Pour se protéger contre des surprises désagréables, il y a deux possibilités différentes. Vous pouvez d'abord protéger "contre l'écriture" vos fichiers importants. Nous verrons plus tard comment cela fonctionne. Vous pouvez d'autre part ajouter à l'instruction ERASE l'option CONFIRM. Cette option peut être abrégée à "C". Si vous entrez maintenant l'instruction suivante:

ERASE *.BAK[C

tous les fichiers backup seront supprimés, mais CP/M vous demandera avant chaque suppression de confirmer que ce fichier doit être également supprimé.

IV.6 Modification des noms de fichier avec REN

Vous avez vu jusqu'à présent que nous nous référons aux fichiers en entrant leur nom de fichier. Il peut cependant arriver qu'un nom ne

vous plaise plus ou que vous vouliez copier un fichier sur une autre disquette sur laquelle un fichier figure déjà sous le même nom. Dans tous ces cas vous pouvez utiliser l'instruction REN ou RENAME sous 3.0 (=renommer) pour rebaptiser vos fichiers. Le format général pour l'instruction RENAME est:

RENAME nouveau=ancien

Vient donc d'abord le nouveau nom du fichier, puis l'ancien. Un signe égale doit être placé entre les deux. Si le changement de nom doit être fait sur le lecteur de disquette annoncé, vous n'êtes pas obligé d'écrire un nom de lecteur de disquette devant le nom de fichier. Si vous essayez de donner comme nouveau nom à un fichier un nom qui existe déjà, CP/M 2.2 vous avertit:

File exists

Le nouveau CP/M est un peu plus serviable puisqu'il vous demande si l'ancien fichier doit être supprimé.

Un changement simple du nom d'un fichier se présente ainsi:

REN nouveau.bas=ancien.bas

Avec RENAME 3.0 vous pouvez aussi utiliser l'étoile et le point d'interrogation. Un exemple simple pourrait se présenter ainsi:

RENAME *.TXT=*.BAK

A la suite de cette instruction, tous les fichiers backup de cette disquette seront transformés en fichiers portant la marque TXT,

quel que soit le contenu de ces fichiers. L'instruction RENAME ne s'occupe en effet absolument pas du contenu des fichiers mais uniquement de leur nom.

IV.7 TYPE

Cette instruction vous permet d'afficher des fichiers sur l'écran. Cela ne fonctionne toutefois qu'avec les fichiers de texte car ceux-ci sont formés par les caractères du jeu de caractères ASCII. Les autres fichiers, avec la marque COM, REL ou autre contiennent des caractères de commande qui perturbent toute sortie sur écran et qui peuvent "planter" l'ordinateur. L'instruction TYPE doit être entrée ainsi:

```
TYPE A:FICHIER.XXX
```

Si vous voulez examiner un fichier sur un autre lecteur de disquette, placez devant un autre nom de lecteur de disquette.

Il y a sous CP/M 3.0 une option pour cette instruction: NO PAGE. Dans ce cas le texte affiché défilera sur l'écran de façon continue alors que la sortie sur écran est normalement interrompue après chaque groupe de 23 lignes pour ne reprendre qu'après que vous ayez entré ENTER.

Si vous avez entré NO PAGE, vous pouvez arrêter la sortie sur écran avec Control S (^S) et la relancer avec Control Q (^Q). Si vous avez auparavant entré encore un Control P (^P), la sortie sur écran sera envoyée sur votre imprimante.

IV.8 Ce que vous savez déjà

* CP/M possède des instructions intégrées ainsi que d'autres qui ne

sont rendues possibles que par un programme figurant sur la disquette

* Les instructions peuvent être entrées avec des paramètres ou des options

* Vous savez comment s'appellent toutes les instructions intégrées, quels sont leurs effets et quelles options ne sont possibles qu'avec les instructions transitoires

V. Instructions transitoires

Vous avez déjà lu un certain nombre de choses sur les instructions intégrées. La véritable force du système d'exploitation se trouve cependant dans les instructions transitoires que vous trouvez comme fichiers COM dans le catalogue. La première partie de ce chapitre traite des instructions de CP/M 2.2, notamment en ce qui concerne STAT.

Dans la seconde partie qui est nettement plus longue, vous découvrirez les nombreuses possibilités de CP/M 3.0 qui vous mettront certainement l'eau à la bouche.

V.1 Affichage d'état avec STAT

L'instruction STAT sert d'une part à afficher l'état du système et d'autre part à modifier l'affectation des périphériques sous CP/M. La forme la plus simple de l'instruction, qui est certainement très souvent utilisée, est:

STAT

suivi d'ENTER. Vous pouvez ainsi faire afficher la place mémoire encore disponible sur la disquette. Cela se présente ainsi:

A:R/W,SPACE:89K

Cette indication signifie que vous avez encore 89 K octets de libre sur la disquette qui peut être lue et sur laquelle vous pouvez écrire.

Ce dernier point est indiqué par R/W ce qui signifie Read/Write

(lire/écrire). A l'opposé existe aussi l'état de disquette R/O (Read Only; lire seulement), lorsque la disquette est dotée d'une protection contre l'écriture. Vous ne pouvez rien écrire alors sur cette disquette mais uniquement en lire son contenu actuel.

La taille d'un fichier ou d'un certain type de fichiers peut cependant être tout aussi intéressante que la place restant sur une disquette. Vous pouvez examiner la taille d'un fichier sur n'importe quel lecteur de disquette en entrant également le nom de ce lecteur de disquette:

STAT B:DONNEES.CMD

Vous obtenez alors à l'écran les renseignements suivants:

RECS BYTES EXT Acc

---

12 2K 1 R/W B:DONNEES.CMD

Bytes Remaining on A: xK

Si vous utilisez l'étoile comme Joker, comme par exemple:

STAT B:*.CMD

vous obtiendrez le nombre qui convient d'affichages d'informations dans le format indiqué plus haut.

Le champ RECS indique combien d'enregistrements comprend le fichier concerné. Dans le champ BYTES figure la taille d'un fichier en unités de 1K et vous pouvez lire sous EXT le nombre de blocs reliés entre eux.

Si vous voulez protéger la disquette entière contre l'écriture, vous pouvez fixer avec STAT l'attribut R/O. R/O est le contraire de l'attribut R/W et permet uniquement de lire des fichiers. Pour rendre effective une telle protection contre l'écriture, entrez:

STAT B:$R/O

Vous pouvez procéder exactement de la même manière si au lieu d'une disquette entière vous ne voulez protéger qu'un fichier isolé ou qu'un groupe de fichiers:

STAT FICHIER.BAS $R/O

Comme l'affectation des différentes zones USER n'est malheureusement pas encore indiquée dans l'interrogation de CP/M 2.2, vous devez utiliser l'instruction STAT pour déterminer dans quelle zone utilisateur vous vous trouvez. Pour prix de votre peine vous voyez cependant également affiché quelles zones utilisateur comportent encore des fichiers et sur quelle zone utilisateur ceux-ci figurent. Vous entrez donc:

STAT USR:

et vous aurez cet affichage:

Active User: 0

Active Files: 0 2 3 5 11 7

Ceci vous indique dans quelle zone utilisateur figurent des

fichiers car si dans une zone ne figure pas au moins un fichier, le numéro de cette zone ne sera pas affiché.

V.2 Instructions transitoires sous CP/M 3.0

Intéressons-nous maintenant à des instructions CP/M 3.0 importantes et souvent utilisées. Il s'agit d'instructions transitoires que vous trouvez toutes dotées d'une marque COM dans votre catalogue. Vous savez déjà comment appeler des programmes: en tapant le nom d'un programme, sans la marque et en appuyant ensuite sur ENTER. Vous avez trois possibilités pour appeler les programmes transitoires. La première, lorsque vous êtes en zone USER 0 et si vous voulez y travailler, la deuxième, lorsque vous avez rendu les programmes accessibles pour toutes les zones USER avec l'instruction SET et la troisième avec l'instruction SETDEF. Nous traiterons ultérieurement plus en détail de cette dernière instruction.

V.3 SET

L'instruction SET effectue sous CP/M 3.0 différentes tâches. Elle est certainement essentiellement utilisée pour fixer divers attributs de fichier. Il s'agit de compléments des fichiers qui ont des effets déterminés. Vous utilisez par exemple l'instruction SET pour rendre vos fichiers CP/M de la zone USER 0 accessibles à toutes les zones USER. Vous pouvez cependant également protéger vos fichiers en disant: "Ce fichier ne doit être que lu" ou "Ce fichier est protégé par un mot de code".

L'instruction SET contient cependant également d'autres options qui influencent le catalogue. Vous pouvez par exemple faire constituer un catalogue dans lequel chaque fichier aura un "tampon avec la date" qui indiquera quand ce fichier a été créé et quand il a été utilisé pour la dernière fois.

Avant de pouvoir créer un catalogue avec des tampons de date (time stamps), vous devez faire tourner le programme INITDIR.COM. Ne croyez pas que les tampons ne constituent qu'un jeu. Vous pouvez utiliser ultérieurement ce système pour sauvegarder automatiquement sur une autre disquette tous les fichiers qui ont été modifiés.

Pour que vous compreniez mieux les multiples possibilités de SET, voici d'abord un aperçu des différentes possibilités:

OPTION	SIGNIFICATION
DIR	rend un fichier système à nouveau visible dans le catalogue normal
SYS	fait d'un fichier un fichier système
RO	fait qu'un fichier ne peut être que lu

RW fait qu'un fichier peut être lu et modifié

ARCHIV=OFF fixe l'attribut ARCHIV sur "non". Cela signifie que ce fichier n'a pas encore été sauvegardé (archivé). Le programme PIP avec l'option [A] peut copier des fichiers avec l'attribut ARCHIV=OFF. Vous entrez l'instruction PIP avec l'étoile pour les noms de fichier et PIP copiera tous les fichiers qui ont été modifiés depuis la dernière opération de copie avec PIP et depuis l'option [A]. Après que PIP ait effectué la copie, il fixe les attributs de fichier sur ARCHIV=ON

ARCHIV=ON fixe l'attribut ARCHIV sur "oui". Cela signifie que ce fichier a été sauvegardé. Normalement, PIP, avec l'option [A], modifie cet attribut après la sauvegarde de fichiers. Vous pouvez également modifier vous-même l'attribut si vous utilisez l'instruction SET.

F1=ON/OFF active ou désactive l'attribut de fichier F1 qui est défini par l'utilisateur.

F2=ON/OFF active ou désactive l'attribut de fichier F2 qui est défini par l'utilisateur.

F3=ON/OFF active ou désactive l'attribut de fichier F3 qui est défini par l'utilisateur.

F4=ON/OFF active ou désactive l'attribut de fichier F4 qui est défini par l'utilisateur.

Nous avons donc vu maintenant ce qui existe et nous allons maintenant voir l'application de ces possibilités. Vous avez quelques programmes importants, par exemple votre programme de texte ou votre banque de données et vous voulez donc travailler sur ces programmes à partir de toutes les zones USER.
Normalement ces programmes figurent, exactement comme les

programmes CP/M, dans la zone utilisateur 0. Afin que vous puissiez toujours accéder à ces programmes ainsi qu'à d'autres fichiers, utilisez l'instruction SET avec une ou plusieurs options pour faire des fichiers en question des fichiers SYSTEME. Cela se présente ainsi:

SET PIP.COM[SYS]

Si vous voulez être sûr que rien ne soit écrit dans le fichier PIP.COM (ou dans tout autre fichier), entrez encore une seconde option:

SET PIP.COM[SYS RO]

PIP.COM devient ainsi un fichier système et en outre un fichier protégé. Si vous voulez maintenant rendre à nouveau librement accessible un fichier ainsi sauvegardé et annuler l'attribut système, entrez:

SET PIP.COM[DIR RW]

L'ordre dans lequel vous entrez les options est sans importance.

V.4 Attributs du lecteur de disquette

Vous avez également la possibilité de définir entièrement un lecteur de disquette de façon à ce qu'on puisse le lire mais non y écrire. Si vous avez fixé un lecteur de disquette sur "RO", ce qui signifie "read only" (lire seulement), aucun fichier ne pourra être supprimé dans ce lecteur de disquette avec ERASE, RENAME ne fonctionnera pas et PIP ne pourra copier aucun fichier sur ce

lecteur de disquette.

Dès que vous entrez un Control C (Tenir enfoncée la touche Control et appuyer sur "C") l'état RO du lecteur de disquette est à nouveau annulé. Si vous entrez:

SET A:[RO]

le lecteur de disquette A est protégé contre l'écriture. Si vous entrez RW (ce qui veut dire read/write = lire/écrire) au lieu de RO, le lecteur de disquette peut à nouveau être pleinement utilisé.

V.5 Les étiquettes

Surtout si vous utilisez beaucoup de disquettes ou si plusieurs utilisateurs travaillent sur le même ordinateur, la possibilité de donner des noms aux disquettes peut se révéler très utile. Utilisez à cet effet également l'instruction SET mais avec l'option "NAME=". Les noms de disquette sont soumis aux mêmes limites que les noms de fichier. Cela signifie que vous pouvez utiliser au maximum huit caractères pour le nom et trois caractères pour la marque.

Pour doter une disquette d'un nom, entrez:

SET B:[NAME=COFI]

Cela pourrait être par exemple le nom d'une disquette comportant votre comptabilité financière. Si vous travaillez déjà sur le lecteur de disquette B:, vous pouvez aussi négliger le nom du lecteur de disquette. Vous ne voyez pas le nom d'une disquette dans le catalogue si vous l'appelez avec DIR. Il vous faut utiliser pour cela l'instruction SHOW, nous vous en dirons plus à ce sujet un peu plus tard, avec en outre l'option "LABEL". Cette option peut être

abrégée à un simple "L" et l'instruction complète se présente ainsi:

SHOW B:[L]

Vous pouvez ici aussi négliger le nom du lecteur de disquette si vous travaillez de toute façon déjà sur B:.

V.6 Mot de passe

Pour que personne ne dérange la belle organisation de vos disquettes, CP/M vous donne la possibilité de convenir d'un mot de code. Vous protégez donc le nom de votre disquette avec un nom de code que vous êtes le seul à connaître.

Dès que vous avez entré un nom de code pour une disquette ou pour un fichier isolé, CP/M demandera avant l'exécution de toute instruction qu'on lui fournisse exactement ce nom de code. C'est pourquoi il vaut mieux que vous vous constituiez un fichier comprenant tous les noms de code, sans quoi vous risquez à la longue d'être pris à votre propre piège. Si vous oubliez un nom de code, vous n'aurez en effet plus aucune possibilité d'accéder à vos données. Donc prudence!

Pour doter votre étiquette d'un nom de code, utilisez l'instruction SET avec l'option "PASSWORD=nom de code" ou dans le cas contraire "PASSWORD=<ENTER>". Entrez le nom de code de la façon suivante:

SET [PASSWORD=SECRET]

ou pour annuler un nom de code:

SET [PASSWORD=(ENTER)]

Encore une fois: pensez bien à noter votre nom de code quelque part. Si vous ne le savez plus, vous n'aurez plus d'accès aux données figurant sur cette disquette.

V.7 PROTECT

Vous ne pouvez pas seulement doter l'étiquette de votre disquette d'un nom de code mais également des fichiers isolés ou même des instructions CP/M. Vous pouvez ainsi vous assurer qu'aucune personne non-autorisée ne viendra jouer avec vos données ou n'aura accès à des informations qui ne la regardent pas.

Avant que vous ne puissiez protéger un fichier isolé ou une instruction, vous devez activer le mode PROTECT. Cela se fait simplement en entrant l'instruction suivante:

SET [PROTECT=ON]

ou inversement:

SET [PROTECT=OFF]

Une fois que vous avez entré cela, vous êtes prêt pour protéger vos fichiers.

V.8 PASSWORD de fichier

Pour placer un fichier sous votre protection personnelle, vous devez procéder fondamentalement de la même façon que précédemment pour la protection du nom de disquette.

Vous pouvez également placer simultanément plusieurs fichiers sous la protection d'un seul nom de code en utilisant la possibilité offerte par l'étoile. Dans ce cas, le même nom de code vaudra pour tous les fichiers pour lesquels la condition est remplie. L'entrée de l'instruction se présente alors ainsi:

```
SET TEXTE.TXT[PASSWORD=nom de code]
```

ou, formulé avec "*":

```
SET *.TXT[PASSWORD=nom de code]
```

Dans le premier exemple, le fichier TEXTE.TXT sera protégé par le "nom de code", dans le second exemple tous les fichiers TXT recevront comme code "nom de code".
On peut encore définir comment les fichiers doivent être protégés. Il existe pour cela les possibilités suivantes:

READ le nom de code est nécessaire pour lire, copier, écrire dans, supprimer ou changer le nom de fichiers

WRITE le nom de code est nécessaire pour écrire dans, supprimer ou changer le nom de fichiers

DELETE le nom de code n'est nécessaire que pour la suppression du fichier

NONE il n'y a pas de nom de code. S'il y a déjà un nom de code, cette option permet de le supprimer

L'entrée doit se faire ainsi:

SET TEXTE.TXT[PROTECT=DELETE]

Vous avez ainsi protégé le fichier TEXTE.TXT contre une suppression inopinée.

V.9 TIME STAMP

Le tampon dateur vous permet de voir si vous utilisez souvent un fichier déterminé, quand un fichier a été créé ou modifié. Il vous permet aussi de gérer vos fichiers de sécurité. Un tampon dateur est comme la pointeuse dans une usine qui permet de conserver toutes les données sur le temps de travail.

Pour pouvoir conserver la date de chaque fichier, vous devez d'abord faire tourner le programme INITDIR. La gestion des dates n'est possible que si vous organisez le catalogue d'une façon particulière. C'est ce que vous permet INITDIR.

Vous disposez de trois options différentes:

CREATE=ON active le marquage de la date pour les fichiers sur un lecteur de disquette déterminé.

ACCESS=ON active le marquage du dernier accès à un fichier. Vous ne pouvez activer que CREATE ou ACCESS. Lorsque ACCESS est voulu sur un lecteur de disquette pour lequel CREATE avait été sélectionné auparavant, CREATE est automatiquement désactivé.

UPDATE=ON active le marquage des fichiers chaque fois qu'ils ont été à nouveau traités. L'affichage dans un catalogue n'a aucun effet sur ce tampon dateur.

Lorsque vous choisissez UPDATE et CREATE comme options, vous devez tenir compte du fait que lors de chaque traitement des fichiers les deux tampons dateurs seront actualisés.

Cela vient du fait que lors du traitement d'un fichier, un nouveau fichier est créé alors que l'ancien devient un fichier BAK (copie de sécurité).

Voici comment vous pouvez mettre en marche la pointeuse de votre ordinateur:

SET [ACCESS=ON]

Pour examiner le résultat de cette option, on entre l'instruction:

DIR[FULL]

et on obtient cela:

Directory for Drive B:

Name Bytes Recs Attributes Prot Update Access

TEXTE.TXT 5K 38 DIR RW NONE 04/01/85 17:31
COFI 20K 152 SYS RO NONE 04/01/85 09:10

Sous le titre ACCESS, vous pouvez voir quand vous avez accédé à un fichier pour la dernière fois. L'instruction pour deux entrées dans le catalogue se présente ainsi:

SET [CREATE=ON,UPDATE=ON]

Un catalogue aurait alors la structure suivante:

Directory for Drive B:

Name Bytes Recs Attributes Prot Update Create

TEXTE.TXT 5K 38 DIR RW NONE 04/17/85 10:00 01/01/85 09:00
COFI 20K 152 SYS RO NONE 04/17/85 16:43 01/01/85 19:21

V.10 SETDEF

Cette instruction vous offre la possibilité de faire rechercher des programmes sur vos disquettes. Vous travaillez par exemple sur le lecteur de disquette B: mais tous les fichiers CP/M ou d'autres programmes se trouvent dans le lecteur de disquette A. SETDEF vous permet de définir pour CP/M une piste de recherche pour qu'il charge les programmes que vous voulez sans que vous ayez chaque fois à entrer le nom du lecteur de disquette sur lequel figurent les programmes.

Si vous entrez simplement:

SETDEF

vous obtenez des informations sur votre piste de recherche actuelle, c'est-à-dire sur le lecteur de disquette utilisé pour les fichiers temporaires et sur les types de fichiers recherchés. Entrez:

SETDEF A:

et CP/M cherchera toujours dans le lecteur de disquette A les fichiers que vous voulez, même si vous travaillez actuellement avec le lecteur de disquette B. Si vous étendez ainsi l'instruction:

SETDEF A:,*

CP/M cherchera les fichiers d'abord dans le lecteur de disquette A: puis dans le lecteur de disquette annoncé. L'étoile est mise pour le lecteur de disquette actuellement annoncé.

Si les fichiers temporaires, comme ceux par exemple que PIP produit, doivent être envoyés sur un lecteur de disquette déterminé, entrez

SETDEF [TEMPORARY=C:]

et les fichiers temporaires seront écrits sur un troisième lecteur de disquette ou également dans la RAM disque.

Il n'est possible de faire rechercher ainsi que des fichiers qui portent une marque COM ou SUB. Sous CP/M 3.0, la recherche concerne normalement les fichiers COM mais cela peut être changé de la façon

suivante:

SETDEF [ORDER=(SUB,COM)]

Les fichiers SUB sont des fichiers qui sont appelés avec l'instruction SUBMIT et qui contiennent à la suite les unes des autres des instructions exécutables. Mais nous y reviendrons plus tard.

V.11 SHOW

Nous en venons maintenant à une instruction qui peut vous donner de nombreuses informations sur la place disponible sur vos disquettes, sur les noms de vos disquettes et sur le nombre de fichiers dans chaque USER. Si vous tapez simplement:

SHOW

vous voyez les attributs de tous les lecteurs de disquette ainsi que la place encore disponible sur chaque lecteur de disquette.

A:RW, Space:101K

B:RW, Space:5,934K

Si vous entrez SHOW suivi d'un nom de lecteur de disquette, vous n'obtiendrez que les indications statistiques concernant ce seul lecteur de disquette.

SHOW vous permet cependant également, comme nous l'avons déjà indiqué plus haut, de faire afficher les étiquettes de vos

disquettes:

SHOW A:[LABEL]

LABEL peut ici être également abrégé à un simple "L".
Vous voyez alors sur l'écran:

Label for drive A:

Directory Passwds Stamp Stamp
Label Reqd Create Update Label Created Label Updated

COFI.COM off off off 04/17/85 11:41 04/17/85 11:41

Avec une option supplémentaire de l'instruction SHOW, vous pouvez faire afficher quelles zones USER sont utilisées sur votre disquette et combien de fichiers correspondent à chaque zone. Le nombre des entrées de catalogue libres est également affiché:

A: Active User: 0
A: Active Files: 0 2 11 12
A: # of files:85 6 1 1

A: Number of free directory entries: 24

Si vous entrez simplement:

SHOW A:[DIR]

seul le nombre des entrées encore libres sera affiché.

V.12 SUBMIT

Vous vous êtes jusqu'ici contenté d'entrer les instructions au clavier. C'est tout à fait logique et c'est bien ainsi mais si vous entrez toujours les mêmes instructions pour obtenir toujours la même chose, cela peut devenir lassant à la longue. CP/M peut vous libérer de cette corvée.

L'instruction SUBMIT vous permet de faire traiter toute une série d'instructions qui se trouvent dans un fichier et qui seront traitées comme des entrées au clavier. Un fichier dans lequel figurent de telles séquences d'instructions est doté de la marque SUB et il peut être lu et exécuté par le programme SUBMIT.

Si votre ordinateur ne possède par exemple pas une horloge intégrée qui soit maintenue en activité en permanence par une batterie, il vous faut entrer la date et l'heure à nouveau chaque fois que vous allumez votre ordinateur. Si vous voulez vous contraindre vous-même à bien faire cette entrée chaque fois afin que les tampons dateurs soient fixés continuellement et correctement, vous devez utiliser le fichier PROFILE.SUB.

Ce fichier a une particularité: il est lu lors de chaque mise en route de CP/M 3.0 et les instructions qu'il contient sont alors exécutées. Si vous prescrivez donc ici que la date et l'heure soient entrées, il ne vous sera pas possible de vous dispenser de cette entrée. Ecrivez ainsi le fichier avec votre traitement de texte:

```
B:DATE SET
```

et baptisez ce fichier PROFILE.SUB. Vous pouvez naturellement indiquer également un autre lecteur de disquette. Cela dépend simplement du lecteur de disquette sur lequel se trouve le fichier DATA.COM. Si vous oubliez la marque SUB, l'instruction SUBMIT n'exécutera rien du tout.

Un fichier SUB peut contenir des instructions CP/M, des instructions SUBMIT imbriquées et des données d'entrée pour un programme ou pour une instruction CP/M. Si vous y réfléchissez un peu, vous constaterez que cela permet de faire beaucoup de choses.

Vous pouvez également travailler dans un fichier SUB avec des indications générales qui seront alors utilisées en fonction de vos entrées. Ces indications générales sont appelées paramètres et elles sont représentées par un signe dollar ($). Vous pouvez utiliser les paramètres $1 à $9.

Supposons que vous écriviez un fichier qui se présente ainsi:

```
ERA $1.BAK
DIR *.$2
```

et que vous appeliez pourquoi pas DIR.SUB. Cette séquence d'instructions dans votre fichier aura pour effet de supprimer d'abord tout fichier portant un nom déterminé avec la marque BAK. Ensuite seront affichés tous les fichiers portant une marque déterminée. Pour arriver à cela, entrez:

```
SUBMIT TEXTE COM
```

Ainsi seront d'abord supprimés tous les fichiers portant le nom TEXTE.BAK puis tous les fichiers portant la marque COM seront affichés. Le fichier SUBMIT une fois traduit se présente en effet ainsi:

```
ERA TEXTE.BAK
DIR *.COM
```

et CP/M exécutera ces instructions exactement comme si elles

avaient été entrées au clavier.

Si vous entrez moins de paramètres qu'il n'en est prévu dans le fichier SUBMIT, les autres paramètres ne seront pas pris en compte. Si vous entrez plus de paramètres qu'il n'en est prévu dans le fichier SUBMIT, les paramètres excédentaires ne seront pas non plus pris en compte. Si vous voulez qu'un signe dollar figure dans une instruction à l'intérieur d'un fichier SUBMIT, entrez simplement deux caractères dollar l'un à la suite de l'autre ($$) et vous obtiendrez ce que vous vouliez.

Un fichier SUBMIT ne peut cependant pas uniquement contenir des instructions sur une seule ligne mais également des entrées d'instructions pour des programmes. Vous pouvez donc appeler un programme avec une ligne d'instruction et transmettre avec la ligne d'instructions suivante des instructions au programme appelé. Un exemple:

```
PIP
<B:=A:*.COM
<
DIR *.COM
```

Vous voyez ici un petit fichier SUBMIT qui contient un élément nouveau. La première ligne appelle PIP.COM, la deuxième entre l'instruction indiquant de copier tous les fichiers COM sur le lecteur de disquette B:. Vous sortez ensuite de PIP et vous voyez s'afficher le catalogue de la disquette. Toutes les instructions qui sont envoyées directement à un programme sont marquées avec le signe "inférieur ou égal" (<). Si vous entrez ce signe sans rien à la suite, comme dans la troisième ligne, cela correspond à un ENTER ce qui, en l'occurrence, met fin à PIP.

L'instruction SUBMIT n'est cependant pas un simple jeu car elle peut recevoir des applications extrêmement utiles. Vous pouvez ainsi par exemple faire imprimer l'un après l'autre autant de fichiers que vous le voulez sans que vous ayez à intervenir entre

temps. Il vous suffit d'écrire ainsi le fichier d'instruction:

```
PIP LST:=FICHIER1
PIP LST:=FICHIER2
PIP LST:=FICHIER3
PIP LST:=FICHIER4
etc...
```

ou bien, ce qui est encore beaucoup plus pratique:

```
PIP LST:=FICHIER?
```

Vous appelez ce fichier TOUTIMPR.SUB et vous entrez, avant de vous retirer:

```
SUBMIT TOUTIMPR (ENTER)
```

Les fichiers peuvent ici figurer également sur différents lecteurs de disquette. Vous devez simplement dans ce cas écrire la lettre du lecteur de disquette devant le nom de fichier et SUBMIT se chargera de retrouver vos fichiers.

Vous pourriez également écrire un fichier d'instruction qui sorte avec DIR la liste de tous les fichiers d'un disque dur à travers toutes les zones USER et qui écrive alors cette sortie sur écran dans un nouveau fichier portant le nom contenu. Vous pourrez alors rechercher dans ce fichier un fichier déterminé avec l'instruction de recherche de votre programme de texte ou faire imprimer la liste des fichiers.

V.13 L'instruction HELP

CP/M 3.0, avec toutes les instructions dont il dispose, n'est plus si facile à assimiler et à maîtriser que ses prédécesseurs. Vous aurez certainement un peu de mal au début à retenir notamment toutes les options possibles. Heureusement, le système d'exploitation vous offre son aide dès que vous appuyez sur une touche. Vous n'avez qu'à crier en anglais "à l'aide!" et vous verrez arriver ce dont vous avez besoin. Par "crier" nous voulions bien sûr dire taper au clavier. Vous entrez donc simplement:

HELP

et vous recevez un menu décrivant les différentes aides possibles. Sélectionnez alors un point de ce menu et vous recevrez des informations plus précises.

HELP UTILITY V1.1

At "HELP>" enter topic [,subtopic]...

EXAMPLE: HELP> DIR EXAMPLES

Topics available:

COMMANDS CNTRLCHARS COPSYS DATE DEVICE DIR

DUMP ED ERASE FILESPEC GENCOM GET

HELP HEXCOM INITDIR LIB LINK MAC

PATCH PIP (COPY) PUT RENAME RMAC SAVE

SET SETDEF SHOW SID SUBMIT TYPE

USER XREF

HELP>

Vous pouvez également appeler directement un des points du menu, en entrant par exemple:

HELP SETDEF

Vous recevrez ainsi directement les informations d'aide. Malheureusement, tout ce qui brille n'est pas or et tout le monde ne comprend pas l'anglais. Bien entendu, nos amis les programmeurs américains partent du principe que le monde entier parle anglais. Cependant les inventeurs de CP/M ont tout de même fini par créer une possibilité pour les non-anglophones d'afficher sur l'écran des informations d'aide dans d'autres langues.

Le programme d'aide se compose des fichiers HELP.COM et HELP.HLP. Pour changer quelque chose dans ce domaine, appelez le fichier HELP.COM et entrez comme option EXTRACT:

HELP [EXTRACT]

Vous pouvez également abréger EXTRACT en un simple "E". Le

programme HELP créera alors à partir du fichier HELP.HLP un nouveau fichier portant le nom HELP.DAT que vous pourrez, avec votre système de traitement de texte, modifier en fonction de vos propres conceptions et traduire dans votre langue.

Pour entrer de nouveaux textes d'aide, vous devez faire attention aux points suivants:

Chaque titre de rubrique doit commencer par trois traits en diagonale (///) suivis d'un numéro. Le numéro indique le niveau d'aide de cette rubrique. Par exemple:

///1DIR (ENTER)

///2OPTIONS (ENTER)

///3PARAMETRES (ENTER)

///4EXEMPLES (ENTER)

Une fois que vous avez tout modifié ou après avoir intégré une aide pour un programme peu utilisé, sauvegardez le fichier et appelez à nouveau HELP.COM, mais cette fois avec l'option CREATE, "C" en abrégé. Un nouveau fichier HELP.HLP sera alors créé qui contiendra vos modifications.

V.14 Ce que vous savez déjà

* Vous avez appris l'importante instruction STAT de CP/M 2.2 et vous pouvez ainsi faire rechercher ou modifier les différentes informations concernant votre système

* Vous connaissez les instructions transitoires de CP/M 3.0

* Vous connaissez maintenant un grand nombre d'options avec

lesquelles vous pouvez modifier ou étendre l'effet des instructions transitoires

* Vous savez que vous pouvez doter les fichiers d'une étiquette

* Vous avez appris comment protéger une disquette entière, un fichier ou certaines instructions CP/M contre un accès non-autorisé

* Vous savez également comment préparer un tampon avec la date et l'heure pour vos fichiers

* Vous connaissez la possibilité de la piste de recherche pour les programmes

* Vous savez maintenant comment les données système d'une disquette peuvent être affichées avec SHOW

* Vous pouvez utiliser le fichier PROFILE.SUB pour automatiser la routine de démarrage de votre ordinateur

VI. Tout sur PIP

Ce n'est déjà plus la première fois que vous entendez parler de cette petite merveille appelée PIP. Afin que vous ne vous imaginiez pas que j'exagère honteusement, voici énumérées rapidement toutes les possibilités de PIP:

* transférer un fichier isolé d'une disquette sur une autre
* transférer un groupe de fichiers d'une disquette sur une autre
* copier un fichier et le doter d'un autre nom
* formater un texte pour l'impression
* raccourcir une ligne trop longue
* imprimer avec une instruction une collection de fichiers
* copier plusieurs fichiers dans un seul
* rechercher une section d'un fichier de texte
* convertir les majuscules en minuscules et inversement
* remettre le huitième bit, dit bit de parité, sur zéro
* doter un fichier de numéros de ligne
* afficher sur l'écran un fichier pendant son transfert
* transférer des fichiers système
* copier des fichiers d'une zone USER dans une autre
* sauvegarder automatiquement les fichiers nouvellement créés ou ayant subi des modifications

Comme vous le voyez ces nombreuses possibilités justifient en tout cas pleinement que nous nous intéressions en détail à ce programme. Lisez ce chapitre ainsi que la section correspondante du manuel plutôt trois fois qu'une afin de bien comprendre tout ce que PIP peut réaliser pour vous.

VI.1 Copie d'une disquette

Le manuel de presque tout programme commence par le conseil de copier d'abord la disquette originale avec PIP sur une disquette neuve, nouvellement formatée, et de ne plus utiliser ensuite que cette copie. C'est ce que nous vous avons également indiqué dans un chapitre précédent, en vous montrant comment cela marche. Comme la logique de PIP est quelque peu différente de la logique normale, voici encore une fois un exemple simple. C'est ainsi que vous pouvez effectuer une copie d'une disquette sur une autre:

```
PIP
*B:TEXTE.TXT=A:TEXTE.TXT
```

Vous obtiendrez ainsi que le fichier TEXTE.TXT de la disquette figurant dans le lecteur de disquette A: soit reproduit sous le même nom sur le lecteur de disquette B:. Votre original reste comme il est mais vous en avez maintenant également une copie dans un autre lecteur de disquette.

Pour copier une disquette entière, entrez:

```
PIP
*B:=A:*.*
```

Vous copiez ainsi tous les fichiers de A: dans B: sans toutefois copier le système d'exploitation. Ce dernier doit en effet être

écrit avec une instruction particulière sur les pistes système de chaque disquette. Cette instruction est:

SYSGEN sous CP/M 2.2 et
COPYSYS sous CP/M 3.0 (sous DISCKIT3!!)

Une fois que vous avez effectué ces deux copies, vous possédez une copie de l'original pleinement utilisable. Vous pouvez introduire votre copie dans le lecteur de disquette et démarrer l'ordinateur. Il existe deux possibilités pour simplifier et accélérer la procédure présentée ci-dessus.

L'instruction PIP peut être dotée de toute une série d'options qui ont des effets extrêmement divers. Une option s'appelle "V", ce qui correspond à "verify". Dès que PIP travaille avec cette option, il vérifie soigneusement après chaque opération de copie si le nouveau fichier correspond bien exactement au fichier d'origine. Les options doivent être ici aussi entrées entre crochets. Pour transférer le fichier TEXTE.TXT et le faire vérifier en même temps, entrez:

```
PIP
*B:=A:TEXTE.TXT[V]
```

C'était le premier pas. Il est cependant ennuyeux d'appeler chaque fois PIP pour entrer ensuite seulement l'instruction dans la seconde ligne. Du point de vue de CP/M, cela n'est d'ailleurs pas du tout nécessaire. Vous pouvez aussi abréger ainsi l'instruction décrite plus haut:

```
PIP B:=A:TEXTE.TXT[V]
```

PIP commencera alors immédiatement son travail qui se conclura

ensuite directement par l'interrogation (A>). Vous avez certainement remarqué que je n'ai pas entré de nom de fichier pour le lecteur de disquette B. Lorsque le nom de fichier ne doit pas être modifié lors de la copie, vous pouvez procéder exactement ainsi. Mais si vous voulez modifier le nom, entrez:

PIP B:TEXTE1.TXT=A:TEXTE.TXT

Vous pouvez ainsi simultanément copier des fichiers et changer leur nom.

Avant cependant que vous ne copiez avec PIP sur une autre disquette, vous devriez vous assurer auparavant qu'il y reste encore suffisamment de place pour le nouveau fichier. PIP transfère en effet le fichier sur une autre disquette et crée sur cette disquette un fichier provisoire que vous pouvez reconnaître au fait qu'il porte le même nom mais avec la marque $$$. Ce n'est qu'une fois qu'il est acquis que le transfert de données a été effectué avec succès que PIP change le nom du fichier provisoire pour lui donner le nom qui convient.

Supposons que vous copiez le fichier TEXTE.TXT sur la disquette B: mais que l'ancienne version de ce fichier figure encore, avec le même nom, sur cette disquette. Que fera PIP? Si la disquette n'offre pas assez de place, vous recevrez un message d'erreur. Cela tient au mode de travail de PIP. Il essaie d'abord d'écrire le fichier à copier sur la disquette, bien que l'ancienne version y figure encore également. D'où le besoin de place. S'il y a assez de place, le nouveau fichier sera transféré, marqué avec $$$, ensuite seulement l'ancien fichier sera supprimé et le nouveau fichier recevra son nom définitif. S'il arrive donc que votre opération de copie ne fonctionne pas, c'est peut-être que vous n'avez plus assez de place pour le nouveau fichier et qu'il vous faut donc d'abord supprimer l'ancien fichier avec ERASE.

Lors de chaque transfert avec PIP sans option spéciale, les attributs de fichier tels que SYS, DIR, RO et RW sont également

transférés avec le fichier. Si vous copiez donc avec PIP un fichier système, la copie sera également un fichier système.

Si vous demandez à PIP de transférer un fichier et qu'il existe déjà sur la disquette objet un fichier du même nom, mais protégé contre l'écriture (RO=Read Only), PIP vous demandera s'il peut supprimer ce fichier.

Répondez "Y" pour oui ou "N" pour non.

VI.2 Copie entre zones utilisateur

Ce que vous avez lu jusqu'ici ne permet d'effectuer de transfert de fichiers qu'à l'intérieur d'une même zone USER. Si vous vous trouvez dans la zone USER 3 et que vous utilisiez PIP pour copier un ou plusieurs fichiers, les copies atterriront sur l'autre disquette également dans la zone USER 3. Pensez-y s'il vous arrive de ne pas retrouver les fichiers que vous avez transférés.

Pour copier d'une disquette sur une autre et en même temps d'une zone USER dans une autre, utilisez l'option "Gn", où "n" est mis pour le numéro de la zone USER. Pour copier le fichier TEXTE.TXT de la zone USER 0 dans le lecteur de disquette B: dans la zone USER 2, entrez:

```
PIP B:[G2]=A:TEXTE.TXT
```

N'oubliez pas que d'autres instructions CP/M comme DIR, ERASE, TYPE, etc... n'ont d'effet que dans la zone USER appelée.

VI.3 Les fichiers de texte et les fichiers non-texte

Jusqu'ici, vous n'avez jamais utilisé PIP que pour amener des fichiers d'une disquette dans une autre. Comme je l'ai déjà indiqué plus haut, vous avez également la possibilité de constituer un fichier global à partir de différents fichiers, et ce en une seule opération.

Avant que vous n'en appreniez plus sur cette intéressante possibilité, nous devons encore vous dire rapidement un mot des différents types de fichier. CP/M distingue fondamentalement deux types de fichier: le fichier de texte et le fichier non-texte. Vous constituez un fichier de texte avec votre programme de traitement de texte ou avec l'éditeur ED intégré dans CP/M (à déconseiller). Vous pouvez utiliser pour cela tous les caractères dont dispose votre clavier. Un fichier non-texte se compose de code binaire et porte en règle générale la marque COM.

Il faut distinguer entre ces deux types de fichier car CP/M reconnaît la fin de fichiers de texte à la présence d'un Control Z (^Z). Tout texte écrit après le ^Z ne sera ni imprimé ni pris en compte d'une manière quelconque. Dans un fichier non-texte par contre, un ^Z est un caractère tout à fait normal et CP/M ne s'en occupe pas.

PIP part normalement toujours du principe que les fichiers à transférer sont des fichiers non-texte et il a ainsi souvent raison. Lorsque toutefois vous voulez réunir en un seul fichier plusieurs fichiers, PIP part du principe qu'il s'agit de fichiers de texte.

VI.4 Réunion de fichiers

La virgule sert à indiquer à PIP qu'il doit réunir plusieurs fichiers entre eux. Pour cette simple raison, aucune virgule ne doit être utilisée dans un nom de fichier, sous peine de perturber PIP totalement.

Voici comment réunir entre eux plusieurs fichiers:

```
PIP TOUT.TXT=PARTIE1.TXT,PARTIE2.TXT,PARTIE3.TXT
```

si tous les fichiers figurent sur le même lecteur de disquette. Si le fichier global doit être écrit dans une autre zone USER, entrez sous CP/M 3.0:

```
PIPTOUT.TXT[G3]=PARTIE1.TXT,PARTIE2.TXT,PARTIE3.TXT
```

Si vous voulez en outre une vérification après la copie, vous devez entrer un [V] après chaque fichier devant devenir une partie du nouveau fichier. La ligne d'instruction se présente donc déjà de façon un peu plus compliquée:

```
PIP TOUT.TXT[G3]=PARTIE1.TXT[V],PARTIE2.TXT[V],
PARTIE3.TXT[V]
```

Cela représente malgré tout toujours moins de travail que de transférer "manuellement" les trois fichiers.

Si vous voulez réunir avec PIP des fichiers non-texte, cela peut poser des problèmes. Comme nous vous l'avons indiqué, chaque fois que PIP rencontre un ^Z, il pense que le fichier est terminé. Le ^Z

peut apparaître assez souvent dans des fichiers non-texte et cela rendrait donc un transfert avec PIP impossible ou très difficile.

Comme les inventeurs de CP/M connaissent ce problème et qu'ils partent du principe que les fichiers COM sont le plus souvent transférés comme des fichiers non-texte, PIP ignore généreusement chaque ^Z dans un fichier COM et le copie ainsi jusqu'à la fin.

Si vous voulez copier des fichiers qui ne sont ni des fichiers de texte ni des fichiers COM, vous devez indiquer le paramètre "O". Comme d'habitude, entre crochets, directement après le nom de fichier.

Vous avez même la possibilité d'écrire encore quelque chose dans chaque fichier pendant l'opération de réunion de plusieurs fichiers. L'instruction de copie doit alors se présenter ainsi:

PIP TOUT.TXT=PARTIE1.TXT,CON:,PARTIE2.TXT,
CON:,PARTIE3.TXT PIP

commence alors à transférer le premier fichier, s'arrête ensuite pour attendre vos entrées au clavier à cause de l'instruction CON:.

Vous devez toutefois entrer à la main les instructions pour ENTER et "nouvelle ligne" (^J) pour que PIP n'efface aucune ligne. En entrant ^Z vous relancez la procédure. Notez cependant que les fautes de frappe ne peuvent pas être corrigées.

VI.5 Numérotation des lignes

Pour les programmeurs, les journalistes et tous ceux qui veulent numéroter leurs textes ligne à ligne, le paramètre "N" de PIP est une possibilité très intéressante. Lorsque vous copiez un fichier avec ce paramètre ou avec son frère "N2", chaque ligne de votre texte recevra son numéro propre.

Si vous faites une copie de la façon suivante:

PIP TEXTENO.TXT=TEXTE.TXT[N]

vos lignes de texte se présenteront ensuite ainsi:

1: Ceci est votre texte

Si vous prenez par contre le paramètre "N2", votre texte se présentera ensuite ainsi:

000001 Ceci est votre texte

Les lignes sont numérotées en ordre croissant ce qui ne peut pas être changé. Donc dans l'ordre: 1, 2, 3, 4, 5...

VI.6 Convertir les lettres

Si au lieu de numéros de ligne vous préférez faire apparaître votre texte en majuscules, vous pouvez utiliser le paramètre "U". Le "U" est mis pour "Uppercase" ce qui signifie majuscules. Vous obtenez l'effet contraire en entrant le paramètre "L" pour "Lowercase" Votre texte ne se composera plus alors que de lettres minuscules.

VI.7 Recherche d'une chaîne de caractères

Il arrive parfois que l'imprimante se mette en grève ou que le papier soit épuisé au beau milieu de l'impression. Vous risquez alors de rencontrer le problème qu'une partie de votre impression

figure encore sur la disquette mais pas sur le papier. Au lieu de faire imprimer à nouveau les 200 pages, utilisez plutôt un autre paramètre de PIP pour faire imprimer le reste du texte sur le papier.

Vous pouvez avec PIP copier une partie de texte en indiquant à PIP à partir de quelle chaîne de caractères et jusqu'à quelle chaîne. Vous marquez la chaîne de début avec un "S" pour "Start" (début) et la chaîne de fin avec un "Q" pour "Quit", ce qui signifie abandonner. Ecrivez un ^Z à la suite de chaque chaîne de caractères et PIP fera le reste. Il examinera votre texte jusqu'à ce qu'il rencontre la chaîne que vous lui avez indiquée. Une telle instruction se présente ainsi:

```
PIP
*PARTTEXT=TEXTE.TXT[Qmot à rechercher^Z]
```

Lorsque vous écrivez cela, PIP commence la copie au début du fichier et s'arrête dès qu'apparaît le mot "mot à rechercher", tel que vous l'avez entré, c'est-à-dire en minuscules.

Si vous écrivez l'instruction ci-dessus dans une ligne, PIP convertira le "mot à rechercher" d'abord en majuscules et commencera alors la recherche. Si votre "mot à rechercher" ne figure pas comme "MOT A RECHERCHER" dans votre fichier, PIP ne trouvera rien et vous enverra un message d'erreur.

Vous pouvez copier une section de texte avec cette instruction:

```
PIP
*PARTTEXT=TEXTE.TXT[Sdébut^ZQfin^Z]
```

Le programme commence alors la copie au mot "début" pour ne s'arrêter qu'à l'apparition du mot "fin".

VI.8 Imprimer plusieurs fichiers à la suite

Vous connaissez jusqu'ici la méthode d'impression d'un fichier qui consiste à entrer ^P et à détourner ensuite avec l'instruction TYPE la sortie sur écran sur l'imprimante. C'est un peu compliqué et cela peut être réalisé de façon plus satisfaisante avec l'aide de PIP. Chaque périphérique porte pour PIP un nom particulier. Le périphérique de sortie est désigné par LST:. Le double point est important pour que PIP puisse distinguer ce nom de périphérique d'un nom de fichier. Si vous voulez faire apparaître votre fichier de texte sur l'imprimante, entrez:

```
PIP LST:=TEXTE.TXT
```

Vous pouvez naturellement utiliser pour cette opération toutes les options qui influencent PIP. Pour faire imprimer par exemple plusieurs fichiers à la suite, entrez:

```
PIP LST:=TEXTE1.TXT,TEXTE2.TXT,TEXTE3.TXT
```

Si vous voulez influer sur le format d'impression de votre texte, utilisez le nom PRN:. Vous aurez alors des lignes de texte numérotées, des pas de tabulation de huit caractères et une nouvelle page commencera toutes les 60 lignes.

A la fin de ce chapitre nous vous donnerons quelques exemples d'utilisations intéressantes de PIP.

VI.9 Sauvegarde automatique de fichiers

Tout aussi intéressante et importante que l'impression des données est bien sûr leur sauvegarde. J'ai déjà attiré votre attention sur ce point et je parle ainsi par expérience. Lors de mes excursions à travers le manuel de PIP j'ai découvert une fonction PIP qui facilite considérablement la sauvegarde. C'est l'option "Archiv", "A" en abrégé.

Surtout si vous travaillez avec un disque dur, vous comprendrez vite l'intérêt de ce paramètre. Chaque fichier sous CP/M 3.0 possède dans sa tête de fichier un emplacement pour un bit appelé "archive flag". Un flag est un témoin qui indique un état déterminé. Chaque fois que vous ouvrez ou modifiez un fichier, ce flag est modifié. Il est par exemple mis sur "1" lors de la modification d'un fichier et est remis sur "0" une fois la sauvegarde des données effectuée.

PIP peut ainsi reconnaître quels fichiers ont été modifiés ou créés depuis la dernière sauvegarde et ne sauvegarder sur disquette que ces fichiers. Vous évitez ainsi d'avoir à sauvegarder chaque fois le contenu entier du disque dur ce qui occasionne une perte importante de temps et d'argent dépensé en disquettes supplémentaires.

Après une longue journée de travail vous pouvez sauvegarder les fichiers modifiés en entrant:

```
PIP B:=A:*.TXT[A]
```

Si vous êtes prudent, vous pouvez ajouter encore l'option "V" et chaque fichier sera vérifié après avoir été copié:

PIP B:=A:*.TXT[AV]

Si vous ne touchez plus sur le disque dur aux originaux des fichiers ainsi sauvegardés, ceux-ci ne seront pas pris en compte lors de la prochaine sauvegarde.

Vous pouvez voir dans le catalogue quels sont les fichiers sauvegardés et lesquels ne le sont pas. Utilisez pour cela l'instruction DIR avec les options FULL ou RW. Vous verrez dans le catalogue les fichiers sauvegardés désignés par "Arcv".

VI.10 Remplacement sans demande de confirmation

Il faut encore tenir compte de certains petits détails lorsque vous travaillez avec PIP. Normalement PIP efface sans hésitation, lors d'une opération de copie, tout fichier de la disquette objet qui porte le même nom que le fichier à copier. Mais s'il s'agit d'un fichier protégé par l'attribut R/O (Read Only), PIP demande si ce fichier doit bien être effacé.

DESTINATION FILE IS R/O, DELETE (Y/N)?

Si vous entrez ici "Y" pour oui, le fichier sur la disquette objet sera effacé et remplacé par le fichier à copier. Si vous entrez par contre "N" pour non, PIP interrompt l'opération de copie. Après "Y" ou "N" vous n'avez par ailleurs pas à appuyer sur ENTER.
Si vous voulez que tous les fichiers soient de toute façon écrits sur la disquette objet et que la demande d'autorisation de suppression soit négligée, utilisez le paramètre "W".

L'instruction se présente alors ainsi:

PIP B:=A:TEXTE1.TXT,TEXTE2.TXT,TEXTE3.TXT[W]

Vous prenez cependant ainsi la pleine responsabilité de la copie et vous ne pourrez pas faire de reproche à PIP.

VI.11 Copie de fichiers système

Si vous essayez de copier un fichier qui possède l'attribut SYSTEME, PIP sera perplexe. Il n'arrivera pas en effet à le trouver. Il faut que vous l'aidiez un peu, avec l'option "R". Vous pouvez naturellement également combiner les deux options. Si des fichiers et des fichiers SYSTEME doivent être copiés sur la disquette objet sans respecter les fichiers protégés, entrez:

```
PIP B:=A:*.COM[RW]
```

Tous les fichiers COM seront ainsi copiés et ils remplaceront les fichiers de même nom qui figuraient éventuellement déjà sur la disquette objet.

VI.12 "Nettoyer" le huitième bit

Le jeu de caractères américain (ASCII) est représenté au moyen de sept bits. La longueur de mot utilisée par l'ordinateur étant habituellement de huit bits, le huitième bit reste disponible pour des tâches spéciales. WordStar l'utilise par exemple pour marquer les espacements pour l'écriture en blocs. MBASIC d'autre part exige que le huitième bit soit libre. Lorsque vous traitez sous WordStar un programme BASIC et que vous sélectionnez par erreur le mode document (D), votre programme BASIC ne pourra pas fonctionner car le huitième bit n'est pas libre. Ce problème est vite résolu avec l'option "Z" de PIP. Il vous suffit de copier le fichier avec PIP et tout est à nouveau en ordre. Ecrivez l'instruction ainsi:

PIP JEU.BAS=JEU.BAS[Z]

Si vous le voulez vous pouvez ajouter à cela l'option "V". Mais n'utilisez pas l'instruction sous cette forme pour copier des fichiers WordStar. Vous perdriez en effet le formatage du texte.

VI.13 Exemples pratiques

Voici maintenant quelques exemples d'applications intéressantes de PIP. Vous avez étudié dans ce chapitre les différents paramètres et vous savez également comment il peuvent être utilisés ensemble.

Le format d'instruction suivant réalise une impression de texte sous une meilleure forme que ^P ou LST: :

PIP LST:TEXTE.TXT[NT8P60]

Le fichier TEXTE.TXT sera ainsi envoyé à l'imprimante, toutes les lignes seront numérotées, des tabulateurs seront disposés toutes les huits colonnes et la longueur de page sera de 60 lignes. Si vous revenez quelques pages en arrière, vous constaterez que ce sont là exactement les valeurs pré-fixées de PRN:. PRN: vous donne donc moins de travail dans ce cas.

Si vous voulez, dans l'exemple ci-dessus, que toutes les lettres apparaissent en minuscules, entrez:

PIP LST:TEXTE.TXT[NT8P60L]

Il suffit donc d'ajouter l'option "L" et ça marche.

Pour rationaliser le travail d'archivage, vous pouvez vous écrire sous CP/M 3.0 un fichier SUBMIT avec le contenu suivant:

```
PIP A:=B:*.*[WAR]
```

Vous appellerez ce fichier ARCHIV.SUB. Les options "WAR" ont pour effet que les fichiers SYSTEME seront également copiés, que les fichiers protégés seront remplacés sans demande de confirmation et que seuls seront copiés les fichiers qui n'ont pas encore été archivés. Si vous employez cette instruction régulièrement, vous ne risquez en fait aucune perte de données. L'appel du programme se fait simplement ainsi:

```
SUBMIT ARCHIV
```

et le reste se fera automatiquement. Vous pouvez encore étendre cette instruction si vous le voulez. "V" entraînera une vérification des fichiers après copie et "E" affichera tout ce qui est copié à l'écran. N'utilisez cependant le "E" que pour des fichiers de texte, sinon vous aurez des problèmes.

D'autres possibilités seraient d'intégrer les instructions DIR et SHOW dans le fichier SUBMIT. Vous verriez alors quels fichiers et combien de fichiers doivent être copiés et combien il reste encore de place sur la disquette.

VI.14 Ce que vous savez déjà

* PIP est un des programmes les plus puissants de CP/M

* Vous pouvez transférer des fichiers isolés sur une autre disquette

* Vous pouvez copier des disquettes entières en une seule fois

* Vous pouvez constituer avec PIP un fichier unique à partir de plusieurs fichiers

* Vous pouvez isoler une partie d'un fichier

* PIP numérote automatiquement pour vous les lignes d'un fichier de texte

* PIP vous permet de sauvegarder automatiquement les fichiers qui ont été dernièrement modifiés

* Vous pouvez convertir les majuscules en minuscules et inversement

* Vous pouvez effectuer des copies d'une zone utilisateur à l'autre

VII. CP/M de l'intérieur

VII.1 Réalisation d'une di quette de sécurité de CP/M

Avant que nous ne nous lancions de toutes nos forces dans CP/M, il faut absolument que vous fassiez une copie de sécurité de votre disquette CP/M. Il peut facilement arriver qu'on supprime par erreur un fichier sur une disquette ou que la disquette soit entièrement détruite de façon incompréhensible. (Et croyez-moi, cela arrive plus souvent qu'on ne le voudrait.)

Vous savez maintenant comment on formate une disquette et comment on copie des fichiers isolés ou une disquette entière. Si vous ne disposez que d'un lecteur de disquette, utilisez l'instruction

DISCCOPY

Cette routine formatera pour vous la disquette objet si nécessaire et vous indiquera à l'écran si vous devez placer dans le lecteur de disquette la disquette source ou la disquette objet.

Vous devez interchanger les disquettes plusieurs fois pendant la procédure de copie. Si vous avez deux lecteurs de disquette, il est donc recommandé d'utiliser l'instruction COPYDISC qui travaille plus vite car sans changement de disquette.

Après que vous aurez réalisé une ou deux copies de sécurité de la disquette CP/M, vous pourrez commencer.

VII.2 Formats de disquette du lecteur AMSTRAD

Le lecteur de disquette AMSTRAD dispose de trois différents formats de disquette: le format CPC standard, le format de données et le format IBM. Ces trois formats ont certains points communs:

- 40 pistes sont formatées (pistes 0 à 39)
- la taille d'un secteur est de 512 octets
- 64 entrées peuvent trouver place dans le catalogue de la disquette

Les formats CPC standard et format de données sont formatés avec 9 secteurs par piste. Ces deux formats diffèrent entre eux uniquement par le fait que dans le format de données les deux pistes supérieures ne reçoivent pas CP/M et sont ainsi disponibles pour le stockage de données ou de programmes. Le format IBM se distingue par le fait que seuls huit secteurs par piste sont formatés. En tant qu'utilisateur de CP/M vous ne devez donc pas utiliser le format de données puisque les pistes importantes 0 et 1 n'y sont pas occupées par CP/M.

Nous nous limiterons au format CPC standard car le format de données n'a rien à voir avec CP/M et le format IBM (ou d'autres) sera décrit ultérieurement. Dans ce format les deux pistes supérieures 0 et 1 sont occupées ainsi:

Piste 0, secteur &41 : secteur boot
Piste 0, secteur &42 : secteur de configuration
Piste 0, secteurs &43 à &47 : inutilisé
Piste 0, secteurs &48, &49 et
Piste 1, secteurs &41 à &49 :CCP et BDOS

(& introduit des nombres hexadécimaux)

CCP signifie Console Command Processor, BDOS est mis pour Basic Disc Operating System.
Vous savez que vous pouvez formater votre disquette avec l'instruction transitoire FORMAT. Sur le CPC, vous ne pouvez cependant formater vos disquettes que sur le lecteur de disquette A:. Comme il y a différents formats, vous devez choisir dans quel format CP/M 2.2 votre disquette doit être formatée. Vous avez les possibilités suivantes:

FORMAT : formate en format CPC standard
FORMAT I : formate en format IBM
FORMAT V : formate en format Vendor
FORMAT D : formate en format de données

Si vous essayez avec un autre paramètre que (S), I, V ou D, FORMAT réclamera ce paramètre. (S) signifie que pour le formatage en format système vous pouvez également négliger le "S".

VII.3 Formats de disquette étrangers

Il est cependant peut-être intéressant pour vous de pouvoir lire ou écrire aussi dans d'autres formats. Il serait très pratique de pouvoir formater ces formats étrangers également sur le CPC. Nous venons de voir que les secteurs comprennent sur l'AMSTRAD 512 octets. Comme le controller du lecteur de disquette peut être programmé à volonté, cette valeur peut être aisément modifiée. Les tailles de secteur envisageables sont 128, 256, 512, 1024, etc... La programmation du controller du lecteur de disquette (PD 765A) est expliquée en détail dans le livre du lecteur de disquette du CPC.

Deux lecteurs de disquette peuvent être connectés sur le controller du lecteur de disquette, y compris des lecteurs de disquette 5.25". La lecture d'un autre format n'a d'ailleurs d'intérêt que pour ceux de nos lecteurs qui possèdent un lecteur de disquette 5.25" car eux

seuls peuvent charger du logiciel CP/M et des données venant d'autres ordinateurs. (L'AMSTRAD CPC est le premier ordinateur avec lecteur de disquette 3".) Mais même les lecteurs qui ne possèdent "que" le lecteur de disquette 3 pouces devraient être surpris des nombreuses possibilités offertes par votre disc controller.

Il faut d'abord remarquer qu'il y a des centaines de formats sous CP/M 2.2 pour les disquettes 5.25". Nous vous présentons ici un tableau qui a été réalisé par la firme Digital Research. Ce tableau indique les différents paramètres des différents formats CP/M et nous permet ainsi de réaliser ces autres formats sur le CPC. Nous nous limiterons ici aux formats pour les disquettes 5.25" simple face car il n'existe plus beaucoup de disquette 8 pouces.

D	B/S	BLS	Cap.	SPT	BSH	BLM	EXM	DSM	DRM	ALO/1	CKS	OFF	PSH	PHM	VR	Sec1	letz	Skew	Nr.
S	128	1	83	18	3	7	0	82	31	80/0	8	3	0	0	0	1	18	5	<1>
S	128	1	83	18	3	7	0	82	63	C0/0	16	3	0	0	0	1	18	4	<2>
S	256	1	92	20	3	7	0	91	63	C0/0	16	3	1	1	0	1	10	1	<3>
S	256	2	92	20	4	F	1	45	63	80/0	16	3	1	1	0	1	10	2	<4>
D	128	1	123	30	3	7	0	122	63	C0/0	16	2	0	0	0	1	30	1	<5>
D	256	1	144	32	3	7	0	142	63	C0/0	16	4	1	1	0	1	16	1	<6>
D	256	1	148	32	3	7	0	147	63	C0/0	16	3	1	1	0	1	16	1	<7>
D	256	1	152	32	3	7	0	151	63	C0/0	16	2	1	1	0	1	16	1	<8>
D	256	1	152	32	3	7	0	151	63	C0/0	16	2	1	1	0	1	16	2	<9>
D	256	1	157	34	3	7	0	156	63	C0/0	16	3	1	1	0	1	17	1	<10>
D	256	2	171	36	4	F	1	84	63	80/0	16	2	1	1	0	1	18	2	<11>
D	256	2	171	36	4	F	1	84	127	C0/0	32	2	1	1	0	0	17	1	<12>
D	512	1	153	32	3	7	0	155	63	C0/0	16	1	2	3	0	1	8	1	<13>
D	512	1	152	32	3	7	0	151	63	C0/0	16	2	2	3	0	1	8	1	<14>
D	512	1	171	36	3	7	0	170	63	C0/0	16	2	2	3	0	1	9	2	<15>
D	512	1	190	40	3	7	0	189	63	C0/0	16	2	2	3	0	1	10	1	<16>
D	512	1	195	40	3	7	0	194	63	F0/0	16	1	2	3	0	0	9	1	<17>
D	512	2	166	36	4	F	1	82	63	80/0	16	3	2	3	0	1	9	1	<18>
D	512	2	190	40	4	F	0	94	63	80/0	16	2	2	3	0	1	10	2	<19>
D	512	2	190	40	4	F	1	94	63	80/0	16	2	2	3	0	1	10	1	<20>
D	512	2	190	40	4	F	1	94	63	80/0	16	2	2	3	0	1	10	2	<21>
D	1024	1	185	40	3	7	0	184	63	C0/0	16	3	3	7	0	1	5	1	<22>
D	1024	2	160	40	4	F	1	78	63	80/0	16	3	3	7	0	1	5	1	<23>

(Source: C'T 6/85, Verlag Heinz Heise Gmbh)

Une table certainement difficile à saisir au premier coup d'oeil. Elle contient toutefois des informations très importantes sans lesquelles nous ne pourrions réussir dans notre entreprise, comme nous allons bientôt pouvoir le constater.

Un point important pour ceux qu'intéresse le format <21>: après leur lecture, les données doivent d'abort être "complémentées", ce

qui peut être obtenu en les XORant avec &FF.

Voici cependant à nouveau la liste des fabricants, pour que vous puissiez mieux identifier les différents formats:

<01>: Xerox
<02>: Lifeboat R2, TRS-80, Mayon
<03>: Eurocom
<04>: Osborne SD
<05>: Superbrain
<06>: MC-CP/M Ecma-70, MC-CP/M+ #7
<07>: Eurocom II Format 2
<08>: Alphatronics DD, NEC PC 8001
<09>: Olympia ETX-II, Philips P-2000
<10>: Spectravideo
<11>: TRS-M4
<12>: Bondwell-12
<13>: IBM CP/M-86
<14>: ADPS
<15>: Dec VT-180
<16>: Rentiki
<17>: Kaypro II
<18>: Olympia Boss A
<19>: SEL Delsy 2000
<20>: Newbrain, Mayon
<21>: Superbrain, HKM-ZDOS
<22>: Osborne DD
<23>: BASF-7120

Les abréviations des première et seconde lignes ont la signification suivante:

D=Density. Vous trouvez dans cette colonne la densité du format: S=Single density (densité simple) et D=Double density. L'apparition d'un format CP/M en quadruple densité n'est maintenant plus qu'une question de temps; les disquettes correspondantes, avec l'étiquette Quad-Density existent déjà.

B/S signifie Bytes per Sector = octets par secteur. Nous avons déjà indiqué que le CPC stocke 512 octets par secteur. 128, 256 ou 1024 octets par secteur sont cependant également possibles.

BLS (BLock Size) désigne la taille d'un bloc en K octets. Avec un format de 128 octets par secteur et une BLS de 1, un bloc regroupe donc 8 secteurs. Sur le CPC, un bloc se compose de 2 secteurs, donc d'un K octets.

Cap. désigne la capacité. Ici est donc indiquée la capacité de stockage du format correspondant en K octets. Le CPC a une capacité de 169 K octets.

SPT (Sectors Per Track) désigne le nombre de secteurs par piste. Le mot secteur ne doit pas ici être compris physiquement mais logiquement. Comme on avait au début des secteurs de seulement 128 octets, on a regroupé ces 128 octets en un enregistrement (record). SPT indique donc combien de ces enregistrements, donc de ces unités de 128 octets, trouvent place sur une piste. Le CPC a neuf secteurs de 512 octets soit 9*512=4608 octets par piste. Si nous divisons 4608 par 128, nous obtenons notre facteur SPT: 4608/128=36.

BSH et BLM sont mis pour Block SHift et BLock Mask. Ces deux indications déterminent la taille d'un bloc. Le controller a besoin de ces indications.

EXM signifie EXtent Mask et fixe le nombre d'entrées dans le catalogue.

DSM fixe le plus grand numéro de bloc possible.

Vous trouverez sous DRM le nombre maximum d'entrées possible dans le catalogue (-1!).

AL0/1 est mis pour la taille du catalogue, codée en binaire. Sur le CPC, deux blocs sont réservés pour le catalogue.

CKS indique de combien de secteurs logiques (enregistrements) se compose le catalogue. 16 sur le CPC.

OFF indique le décalage de piste. Ici figure le nombre de pistes réservées pour CP/M.

PSH et PHM fournissent des informations sur le "blocking" et le "deblocking" de secteurs physiques en secteurs logiques.

Sec1 donne le numéro du premier secteur, letz le numéro du dernier secteur d'une piste. Sur le CPC, le premier secteur est le secteur No 1, le dernier le numéro 9.

La dernière indication, le facteur skew, n'a d'intérêt que pour les initiés.

Maintenant que nous disposons de toutes ces informations, nous allons essayer de voir quelles informations peuvent être communiquées au CPC ou plutôt au controller PD 765A. Examinons pour cela la mémoire système du lecteur de disquette du CPC:

Vous trouvez ces indications dans la RAM système à l'adresse &A890 - &A8A5.

A890,A891	SPT	Enregistrements par piste (36)
A892	BSH	Block SHift (3)
A893	BLM	BLock Mask (7)
A894	EXM	EXtend Mask (0)
A895, A896	DSM	Numéro de bloc maximal (170)
A897, A898	DRM	Entrées maximum dans le catalogue -1 (63)
A899, A89A	AL0/1	Taille du catalogue (C00H), codée en binaire, correspond à deux blocs
A89B, A89C	CKS	Nombre d'entrées à vérifier dans le catalogue

		(&0010) 16 entrées
A89D, A89E	OFF	Décalage de piste (2) Pistes système occupées
A89F	FSC	Premier secteur de chaque piste (041h)
A8A0	PST	Secteurs physiques par piste (9)
A8A1	GPS	Longueur GAP3 pour Read/Write secteur (2Ah)
A8A2	GPT	Longueur GAP3 lors du formatage de piste (52h)
A8A3	FLB	Octet de remplissage lors du formatage de piste (E5h)
A8A4	BPS	Octets par secteur (2) correspond à 512 octets
A8A5	RPS	Nombre d'enregistrements par secteur (4)

Les valeurs entre parenthèses sont les valeurs par défaut, c'est-à-dire les valeurs pré-fixées.

Le format <13>, le format IBM, est le seul que le CPC mette automatiquement à votre disposition.

La réalisation d'un autre format ne pose en fait aucun problème: vous voyez bien qu'on peut tout indiquer au controller. La seule difficulté qu'il nous reste à résoudre est le choix de la longueur du GAP3. Le GAP3 est un vide qui est utilisé sur les disquettes pour absorber de petites variations de la vitesse du lecteur de disquette car il est à peu près certain que le lecteur de disquette n'a pas exactement la même vitesse lors de l'écriture d'un secteur que lors du formatage de ce secteur. Le danger existe alors, avec une vitesse moindre, qu'on écrive sur le prochain secteur (physique). Les vides GAP3 servent donc ici de tampon. Plus ce GAP3 sera grand et moins on pourra mettre de secteurs sur une piste. Si on choisit cependant un GAP3 trop petit, le danger augmente qu'on écrive sur le secteur suivant. Un GAP3 est une suite de codes &4E mais n'importe quel autre caractère pourrait être envisagé également.

Essayons donc de réaliser sur notre AMSTRAD le format OSBORNE <22>.

Nous pouvons déduire de notre tableau les faits importants suivants:

1024 octets par secteur, 5 secteurs par piste, secteurs No 1 à 5. Ces informations ainsi que les autres doivent donc être écrites dans la mémoire système de l'ordinateur. Il est essentiel, et cela a pris un certain temps lors du développement du programme machine, que vous stockiez vraiment 16 bits si des valeurs de 16 bits sont prévues. Sinon en effet toutes les informations suivantes se décaleront dans la mémoire système du controller du lecteur de disquette et vous n'obtiendrez certainement pas le résultat escompté.

Après que vous ayez pourvu les adresses de mémoire système &A890-&A8A5, les conditions de base pour un autre format sont réunies. Comme vous pouvez le voir d'après le programme machine qui vous est donné dans les pages suivantes, on peut facilement copier avec un tableau et l'instruction LDIR les données dont on dispose dans la mémoire système. Vous pouvez utiliser vos routines système comme auparavant et elles fonctionnent sans problème comme le montre notre programme d'exemple. Les 40 pistes sont d'abord formatées avec 5 secteurs par piste. Une fois que cela est réalisé, on écrit dans tous les secteurs pour les tester. Ce test doit montrer si tous les secteurs sont effectivement disponibles. Il se pourrait en effet que le GAP3 ait été choisi trop petit ou trop grand ou que quelque chose d'autre n'ait pas fonctionné. Si vous regardez à quelle vitesse le lecteur de disquette formate les 40 pistes et à quelle vitesse il écrit à nouveau sur ces pistes, vous mesurerez la puissance de ce lecteur de disquette.

Mais voici tout d'abord le programme assembleur de formatage en format OSBORNE

```
10
20 ;Formatage en format Osborne, JS 9/6/85
30 ;
40 ;
50          ORG #7000 ;ADRESSE DE DEPART
60 ;
70 ;Initialise FDC avec les nouvelles valeurs:
80 ;
```

```
90                  LD          BC,22       ;22 VALEURS
100         LD      HL,FDC      ;TABLE DES NOUVELLES VALEURS
110         LD      DE,#A890    ;MEMOIRE SYSTEME DU LECTEUR
120         LDIR
130 ;
140         LD      E,0         ;LECTEUR
150         LD      D,0         ;PISTE
160         LD      B,40        ;40 PISTES
170 L1:     PUSH    DE          ;SAUVER DE
180         PUSH    BC
190         LD      HL,FTAB     ;TABLE POUR LE FORMATAGE
200         LD      A,D         ;PISTE=>D
210         LD      B,5         ;5 SECTEURS
220 LO:     LD      (HL),A      ;SAUVE PISTE
230         INC     HL
240         INC     HL
250         INC     HL
260         INC     HL          ;POINTEURSURLEPROCHAINSECTEUR
270         DJNZ    LO          ;PROCHAIN SECTEUR
280         LD      C,#1        ;PREMIER SECTEUR
290         LD      HL,FTAB     ;TABLE POUR LE FORMATAGE
300         RST     #18
310         DEFW    FORMAT      ;PISTE FORMATEE
320         POP     BC
330         POP     DE
340         INC     D           ;PISTE SUIVANTE
350         DJNZ    L1
360 ;
370         JP      TESTE       ;VERIFIE TOUS LES SECTEURS
380 ;
390 ;
400 FORMAT: DEFW    #C652       ;ADRESSE #C652
410         DEFB    7           ;DANS LA ROM DISQUETTE
420 ;
430 FTAB:   EQU     $
440 ;
450         DEFB    0           ;PISTE
460         DEFB    0           ;ADRESSE DE TETE DU LECTEUR
470         DEFB    1           ;SECTEUR
```

```
480          DEFB      3            ;3=1024 OCTETS
490          DEFB      0,0,3,3      ;SECTEUR 3
500          DEFB      0,0,5,3      ;SECTEUR 5
510          DEFB      0,0,2,3      ;SECTEUR 2
520          DEFB      0,0,4,3      ;SECTEUR 4
530 ;
540 ;
550 ;TABLE DES VALEURS DU CONTROLLER
560 ;
570 FDC:     DEFB      40,0,3,7,0,184,0,63,0,#C0,0,16
580          DEFB      0,3,0,1,5,42,50,#E5,3,8
590 ;
600 ;ECRITURE-TEST SUR TOUS LES SECTEURS
610 ;
620          ORG       #7100        ;NOUVELLE ADRESSE D'ENTREE !!
630 ;
640 TESTE:   CALL      #BB06        ;ATTENDRE FRAPPE D'UNE TOUCHE
650          LD        E,0          ;LECTEUR
660          LD        D,0          ;PISTE
670 LL1:     LD        C,5          ;SECTEUR
680          LD        B,5          ;COMPTEUR
690          LD        HL,#5000     ;BUFFER QUI EST SAUVEGARDE
700 LL2:     RST       #18
710          DEFW      WRITE        ;ECRIT UN SECTEUR
720          JR        NC,ERREUR    ;ERREUR APPARUE
730          DEC       C            ;PROCHAIN SECTEUR
740          DJNZ      LL2
750          INC       D            ;PISTE SUIVANTE
760          LD        A,D
770          CP        40           ;TOUTES LES 40 PISTES??
780          JR        NZ,LL1       ;PAS ENCORE TOUTES
790          RET                    ;RETOUR
800 ;
810 WRITE:   DEFW      #C64E
820          DEFB      7            ;ADRESSE#C64EDANSLAROMDISQUETTE
830 ;
840 ERREUR:  EQU       $            ;ANNONCER ERREUR PAR UN BIP
850          LD        A,7          ;BIP
860          CALL      #BB5A        ;SORTIR
```

```
870         RET                     ;REND LA MAIN
880 ;
890 ;
900         ORG     #7200           ;NOUVELLE ADRESSE D'ENTREE
910 ;
920         LD      E,0             ;LECTEUR
930         LD      D,1             ;PISTE
940         LD      C,2             ;SECTEUR
950         LD      HL,#5000        ;BUFFER
960         RST     #18
970         DEFW    READ            ;LIT SECTEUR
980         RET                     ;FIN DE LA PROCEDURE
990 ;
1000 READ:  DEFW    #C666
1010        DEFB    7               ;ADRESSE #C666 DANS LA ROM DISQUETTE
1020 ;
```

Nous ne vous présentons pas ici de programme de chargement BASIC car cela n'aurait certainement pas d'intérêt. Si vous vouliez en effet réaliser un autre format que le format OSBORNE, le programme de chargement tout entier serait inutilisable. De toute façon, les lecteurs qui parmi vous veulent lire sur leur CPC un autre format doivent disposer de connaissances en langage machine car "votre" CP/M ne traite pas aussi facilement le nouveau format. Vous devez charger les données par programme machine, en vous aidant des routines BDOS.

Avec une modification appropriée de la table FDC (lignes 570-580) tous les formats que nous vous avons présentés sont possibles. Veillez simplement à ce que les valeurs soient placées dans l'ordre de la mémoire système. Toutes les valeurs apparaissant dans la table mais pas dans la mémoire système peuvent être négligées.

Pour vous rendre compte à quelle vitesse 1024 octets sont chargés, essayez de faire le test suivant: comme on écrit sur toutes les pistes après le formatage, il faut bien que les données écrites viennent d'une zone quelconque de la mémoire. Dans l'exemple suivant, nous avons choisi pour cela la zone de mémoire &5000-

&5400. Entrez les lignes BASIC suivantes avant le formatage:

```
DEFINT I
FOR I=&5000 TO &5400
  POKE I,I AND &FF
NEXT
CALL &7000 (pour lancer le formatage)
```

La zone de mémoire &5000-&5400 est maintenant occupée par des données. (On compte de 0 à 255 et on recommence ensuite à 0.) Cette mémoire a été également copiée sur la disquette (200 fois) après l'appel CALL &7000.

A l'adresse &7200 se trouve encore une petite routine pour lire un secteur de 1024 octets. Mais avant que vous n'appeliez cette routine, il serait préférable que vous vidiez la mémoire car sinon vous ne pourrez pas contrôler si la mémoire a bien été pourvue de données venant du lecteur de disquette:

```
FOR I=&5000 TO &5400
  POKE I,0
NEXT
```

Une fois que vous avez vidé la mémoire, vous pouvez essayer de charger un bloc quelconque - de toute façon, ils sont tous identiques:

```
CALL &7200
```

et le secteur sera lu à toute vitesse. Vous pouvez vous en persuader en faisant afficher cette zone de la mémoire:

```
FOR I=&5000 TO &5400
  PRINT PEEK(I);
NEXT
```

Vous devriez obtenir comme résultat:

1 2 3 4 5 6 7 8 9 10 11 12 13 ....

Cela ne constitue pas seulement un exemple montrant la vitesse à laquelle sont écrites les données mais également un test pour vérifier si ce sont vraiment 1024 octets qui ont été stockés et chargés.

Encore un petit conseil qui peut se révéler très important dans certaines circonstances: comme sur le CPC on ne charge de façon standard que des secteurs de 512 octets, vous ne disposez également que d'un buffer de 512 octets pour la lecture des secteurs. Mais comme lorsque vous lisez 1024 octets ce buffer est en fait "automatiquement" étendu à 1024 octets, il peut arriver que des données ou des sections de programme importantes soient effacées. Cela ne représente cependant pas un gros problème puisque vous pouvez placer le buffer de secteur où vous le voulez. Le vecteur pour le buffer de secteur se trouve à l'adresse:

&BE62, &BE63

Vous y arriverez donc par exemple en ajoutant les lignes suivantes au programme assembleur:

```
135             LD   HL,SEKBUF
137             LD   (#BE62),HL   ;DEPLACER LE BUFFER
1030 SEKBUF:DS  1024              ;BUFFER DE 1024 OCTETS
```

VII.4 La disquette CP/M

Mais revenons maintenant au format que nous n'avons pas à simuler péniblement. Avec l'AMSTRAD CPC 464 et CPC 664 est livrée une disquette sur laquelle figurent LOGO de Digital Research ainsi que CP/M 2.2. Nous allons tout d'abord voir de plus près tout ce qu'il y a sur la disquette CP/M - il n'y a en effet pas que les fichiers d'instructions CP/M même si ceux-ci représentent de loin la partie la plus importante:

```
A: MOVCPM   COM : PIP      COM : SUBMIT   COM : XSUB     COM
A: ED       COM : ASM      COM : DDT      COM : LOAD     COM
A: STAT     COM : DUMP     COM : DUMP     ASM : AMSDOS   COM
A: FILECOPY COM : SYSGEN   COM : BOOTGEN  COM : COPYDISC COM
A: CHKDISC  COM : DISCCOPY COM : DISCCHK  COM : SETUP    COM
A: FORMAT   COM : CSAVE    COM : CLOAD    COM : EX1      BAS
A: EX2      BAS : ROINTIME DEM
```

Sur la disquette se trouvent par exemple des fichiers de démonstration pour le BASIC sous AMSDOS que nous laisserons volontairement de côté. Si nous nous limitons aux instructions CP/M, nous constatons qu'à côté des instructions standard CP/M 2.2 figurent encore quelques autres fichiers COM, comme par exemple CLOAD.COM et CSAVE.COM. Ces deux fichiers d'instruction servent à copier des fichiers CP/M de la disquette sur le lecteur de cassette (CSAVE) ou de la cassette sur la disquette (CLOAD). Si vous voulez par exemple stocker le fichier FILECOPY.COM sur cassette, entrez simplement l'instruction suivante dans votre ordinateur:

CSAVE FILECOPY.COM (ENTER)

C'est tout. Sur votre écran apparaît alors:

CSAVE V2.0

Press REC and PLAY then any key:
Saving FILECOPY.COM block x

Lorsque vous avez appuyé sur les deux touches correspondantes de votre lecteur de cassette, vous pouvez frapper une touche quelconque et l'opération de copie commencera. L'indication "block x" vous indique quel bloc est en train d'être stocké sur la cassette. Une fois que tout a été stocké sur la cassette, apparaît sur le moniteur:

CSAVE V2.0

Il peut être très intéressant de stocker sur cassette certains fichiers que vous pouvez relire ensuite avec l'instruction CLOAD car les cassettes constituent un moyen de stockage très très bon marché, au contraire des disquettes 3 pouces qui sont (encore) très chères. Vous pouvez donc archiver ainsi certains fichiers. Et d'ailleurs, pourquoi pas? Sur les gros ordinateurs, on ne fait pas autre chose et les disques sont régulièrement copiés sur les bandes magnétiques bien moins onéreuses.

VII.5 L'assembleur ASM livré avec le lecteur

Sur la disquette CP/M qui accompagne tout lecteur de disquette DDI-1 ainsi bien sûr que le CPC 664 et le CPC 6128, se trouve un assembleur. Vous pouvez appeler cet assembleur avec ASM. Mais attention! Ne vous réjouissez pas trop vite. Comme vous le savez peut-être, l'AMSTRAD possède comme unité centrale un processeur Z80. Vous avez peut-être déjà appris la programmation en langage machine grâce à un travail acharné. Mais cet assembleur n'est pas un assembleur Z80 mais un assembleur 8080. Pour ceux d'entre vous qui ne savent pas encore exactement ce que cela veut dire: le 8080

est plus ancien que le Z80 dont il est en quelque sorte le père. Le Z80 comprend les programmes que son père comprend et il possède même un vocabulaire beaucoup plus étendu que son père mais il utilise un autre code de mnémoniques. Vous avez bien lu. Le code mnémonique est le mode d'écriture que les programmeurs utilisent pour coder leurs programmes machine. Vous savez certainement que les programmes machine ne se composent que de nombres. Par exemple, le processeur Z80 comprend l'instruction &41 et il peut l'exécuter. Pour un homme, il est toutefois difficile de ne penser qu'en nombres. C'est pourquoi on a développé les codes mnémoniques qui remplacent temporairement les codes machine. Sur le Z80, le code mnémonique pour l'instruction &41 se présente ainsi:

LD B,C

Le programmeur en langage machine voit maintenant tout de suite de quoi il s'agit et cela simplifie considérablement le travail quand on n'a pas à consulter en permanence des tableaux pour savoir par exemple ce que le code &41 a comme effets.

L'assembleur a pour fonction de traduire ces codes mnémoniques que l'homme comprend "si bien" en codes compréhensibles par la machine. Le processeur ne peut en effet rien tirer du code mnémonique LD B,C. L'assembleur intervient donc ici comme interprète ou interface pour traduire le langage des hommes en langage machine. Mais l'assembleur aide encore l'homme dans son travail par d'autres moyens sur lesquels je reviendrai brièvement un peu plus loin.

Vous savez donc maintenant ce qu'est un code mnémonique et que le vieux processeur 8080 est le père du Z80. Vous savez encore que le Z80 maîtrise plus d'instructions que le 8080. Tous les programmes CP/M doivent cependant être programmés en code 8080 car il y a aussi des ordinateurs CP/M qui possèdent un 8080. Cela signifie donc qu'un programmeur Z80 doit se limiter aux instructions que le 8080 comprend lui aussi (et qu'il doit donc ainsi renoncer à un certain nombre d'instructions vraiment remarquables du Z80). Mais ce n'est malheureusement pas tout car on a en outre également

modifié le code mnémonique lors de l'évolution qui a mené du 8080 au développement du Z80. Souvenons-nous de notre instruction &41. La reconnaissance et l'exécution de cette instruction a les mêmes effets sur les deux processeurs: le contenu du registre C est copié dans le registre B. Le code mnémonique se présente ainsi sur le Z80:

LD B,C

alors que la même (!!) instruction se présente ainsi sur le 8080:

MOV B,C

Vous voyez donc certainement déjà où est le problème: vos connaissances du Z80 ne vous serviraient pas de grand chose si vous vouliez programmer un 8080, même si vous vous limitiez aux instructions que le 8080 peut aussi comprendre. Donc: vous ne pouvez alimenter l'assembleur ASM qu'avec des codes qu'un assembleur 8080 peut comprendre.

Nous allons cependant examiner de plus près cet assembleur qui nous permet de programmer nos propres fichiers COM.

VII.6 L'emploi de l'assembleur ASM

Dans la programmation en langage machine, il arrive fréquemment que l'on doive se référer à des cases mémoire déterminées, par exemple pour charger ou sauver des registres. Il arrive également très souvent que l'on saute à une adresse mémoire déterminée. Supposez par exemple que vous ayez écrit un programme qui commence à l'adresse &5000 et que vous vouliez maintenant le décaler vers l'adresse &4000. Vous devriez modifier toutes les adresses de saut autres que les adresses de saut relatif. C'est alors qu'un

programmeur peut bien craquer. Pour faciliter ce travail on a créé dans les assembleurs la possibilité de définir des étiquettes (labels) qui seront remplacées par leur valeur lors de l'assemblage (c'est-à-dire lors de la traduction du code mnémonique en langage machine). Désormais, ce n'est donc plus un problème d'insérer une instruction supplémentaire à n'importe quel endroit - l'assembleur prend en charge à votre place tous les travaux désagréables.

Voulez-vous savoir comment se présente un tel programme d'assembleur? Pour vous cela ne pose aucun problème. On a eu l'amabilité de placer sur votre disquette CP/M un programme d'assembleur avec lequel nous allons maintenant nous exercer à la manipulation de l'assembleur. Examinez encore une fois attentivement le catalogue de votre disquette CP/M. Nous avons un fichier appelé DUMP.COM; c'est, comme nous le savons déjà, un fichier COM que nous pouvons lancer sous CP/M en en tapant simplement le nom:

DUMP <nom de fichier>

serait ici la syntaxe. Dump vous fournit un affichage du fichier fourni en paramètre, en format HEX (HEX DUMP). Essayez donc:

DUMP FILECOPY.COM

Sur l'écran apparaît l'hex-dump du fichier COM Filecopy et donc en quelque sorte de ce programme. Vous pouvez interrompre la sortie avec <CTRL>/S et la relancer avec <CTRL>/Q. <CTRL>/C vous permet d'arrêter la sortie.

DUMP.ASM est le programme machine de DUMP, tel qu'il a été écrit en code par le programmeur. Vous pouvez faire sortir ce fichier sur l'écran (ou l'imprimante). Entrez pour cela:

TYPE DUMP.ASM

La tête du programme vous indique que la firme DIGITAL RESEARCH possède les droits pour ce programme. Ce n'est d'ailleurs pas étonnant car c'est justement cette société qui a développé CP/M. Vous voyez ici encore une autre différence essentielle entre l'assembleur et le langage machine: vous pouvez ajouter des commentaires. Ces commentaires commencent par un point-virgule pour que l'assembleur sache à partir de quelle position il doit arrêter la traduction en code. Sinon l'assembleur essaierait naturellement de comprendre ce commentaire pour pouvoir le traduire et il ne pourrait que hocher la tête et dire: il y a ici quelque chose qui ne va pas!! Même si vous trouvez ce programme très long, croyez-moi, il est en fait très court. Quand vous pensez en effet que les systèmes d'exploitation sont également codés en assembleur:

Il y a des systèmes d'exploitation qui occupent à l'impression plus de 2000 pages. Même des programmes "plus petits" comme par exemple le traitement de texte TEXTOMAT sur votre AMSTRAD comptent encore facilement 400 à 500 pages. Vous voyez que la programmation en langage machine représente un énorme travail. Mais nous allons maintenant assembler ce programme assembleur pour que vous puissiez à l'avenir utiliser vous-même l'assembleur. Faites attention à ce que votre disquette de travail ne soit pas protégée pour le moment car il nous faut également écrire sur la disquette sur laquelle se trouve le code source. vous avez certainement déjà réalisé une copie de sécurité de votre disquette CP/M?! (Si ce n'est pas encore le cas, c'est le moment ou jamais!) Assemblons maintenant "notre" programme machine:

ASM DUMP

Vous voyez ici une caractéristique importante de l'assembleur: vous n'avez pas besoin d'entrer l'extension, c'est-à-dire que vous n'êtes pas obligé d'indiquer DUMP.ASM car l'assembleur ajoute de lui-même cette extension .ASM. Sur votre écran apparaît:

```
CP/M ASSEMBLER - VER 2.0
0257
002H USE FACTOR
END OF ASSEMBLY

A>
```

Il est rapide notre assembleur, pas vrai? Il crée maintenant deux fichiers:

1) un protocole sous le nom de DUMP.PRN
2) un hex-dump sous le nom de DUMP.HEX

Examinez d'abord le protocole. Ici est à nouveau listé le programme assembleur avec toutes les informations ajoutées par l'assembleur. Dans la première colonne figure toujours l'adresse à laquelle il est en train de créer du code, sauf s'il est en train de définir une étiquette, auquel cas figurera dans la première colonne la valeur ainsi définie de l'étiquette. Le programme commence donc à l'adresse &0100 et finit en &0257. Nous savons donc maintenant à quoi correspondait le nombre sorti par l'assembleur (voir plus haut). Presque tous les programmes CP/M commencent à l'adresse &0100. Après que vous ayez bien examiné le protocole ou que vous l'ayez même recopié sur le papier, examinez maintenant le second fichier que notre courageux assembleur a réalisé. Entrez pour cela (comme vous le savez certainement maintenant):

TYPE DUMP.HEX

Vous voyez maintenant sur l'écran une série de nombres qui ont été produits par l'assembleur. Il s'agit de codes programme. Quelques autres informations sont également affichées, comme l'adresse à laquelle devra figurer le code plus tard, etc... Ce fichier est déjà beaucoup plus court que notre fichier DUMP.PRN, n'est-ce pas?

Mais nous ne pouvons toujours pas lancer ce programme machine car ce n'est pas encore un fichier COM. Il y a maintenant encore un autre programme qui se charge de ce travail. Il constitue à partir d'un fichier HEX un fichier COM, c'est-à-dire un programme que CP/M peut appeler et exécuter. Ce programme s'appelle LOAD. Appelons ce programme:

LOAD DUMP

Vous voyez que nous n'avons pas ici non plus besoin ou plutôt que nous ne devons pas entrer d'extension (marque). LOAD ajoute automatiquement l'extension ".HEX". Sur l'écran apparaît:

FIRST ADDRESS 0100
LAST ADDRESS 0212
BYTES READ 0113
RECORDS WRITTEN 03

C'est maintenant enfin terminé: vous avez créé un fichier COM, le fichier DUMP.COM que nous pouvons maintenant appeler sous CP/M en entrant simplement:

DUMP

Mais il n'est pas si facile de programmer des fichiers COM car on doit très bien connaître le CCP et le BDOS puisqu'il faut utiliser ces routines.

Nous allons cependant vous livrer encore quelques informations sur l'assembleur afin que vous puissiez l'utiliser correctement (si vous le souhaitez).

Les programmes source pour l'assembleur doivent présenter la

syntaxe suivante:

<Numérodeligne><Etiquette>:<Instruction><Argument>;<Commentaire>

Le <numéro de ligne> est facultatif, c'est-à-dire que vous pouvez vous en dispenser (comme dans notre fichier d'exemple). Il est de toute façon ignoré par l'assembleur et il n'a été prévu que parce que certains éditeurs ajoutent ces numéros de lignes automatiquement, comme par exemple l'éditeur ED sur la disquette CP/M. l'<Etiquette> n'est naturellement pas non plus obligatoire mais elle peut figurer à cet endroit. Dans une ligne qui ne consiste pas simplement en un commentaire doivent obligatoirement figurer l'<Instruction> et l'<Argument>. A cela s'ajoute ensuite éventuellement le ;<Commentaire>.

L'assembleur convertit avant le codage toutes les minuscules en majuscules comme vous en avez du reste déjà l'habitude sous CP/M. Cela facilite déjà beaucoup la programmation. Exclues de cette règle sont cependant les lettres minuscules qui figurent entre apostrophes et qui représentent un texte défini, comme par exemple l'instruction:

```
DB 'File Dump Version 1.4$'
```

Les différents composants de la ligne assembleur doivent être au moins séparés par un espace. Il est usuel de faire commencer l'étiquette complètement à gauche et d'appeler ensuite au moyen d'un tabulateur, dans le traitement de texte avec lequel vous écrivez le code source assembleur, les différents composants. Cela n'est certes pas indispensable, comme nous l'avons déjà indiqué, mais cela augmente considérablement la lisibilité du programme car les éléments de même nature figurent ainsi les uns sous les autres. Exemple:

```
DISKR:     ;READ DISK FILE RECORD
           PUSH H! PUSH D! PUSH B
           LXI  D,FCB
           MVI  C,READF
           CALL BDOS
           POP B! POP D! POP H
           RET
```

Vous trouverez cette section dans notre programme DUMP. DISKR est une étiquette, marquée par une double point. Viennent ensuite l'<Instruction> et l'<Argument> qui sont toujours mélangés. Dans une ligne nous avons même un commentaire, précédé d'un point-virgule ";".

Vous voyez dans cette petite routine une autre particularité de l'assembleur ASM: les différentes instructions sont séparées entre elles pas le point d'exclamation.

On peut avec ASM marquer les commentaires par un point-virgule ou définir une ligne entière de commentaire en commençant par une étoile dans la première colonne. Cela a été introduit par les développeurs de ASM pour conserver une certaine compatibilité avec les assembleurs 8080 existants. On peut donc ainsi doter les différentes routines d'en-têtes, comme par exemple:



```
*************************************************************
***        Read Disk File Record                          ***
*************************************************************
```

Faites attention dans l'emploi des étiquettes à ne pas utiliser de mots-clés. Les mots-clés sont par exemple les codes mnémoniques du processeur 8080, donc des mots comme MVI, MOV, STA etc...

Avec l'assembleur ASM, vous pouvez aussi, comme avec beaucoup d'autres assembleurs, déterminer la position actuelle du Program Counter pendant l'opération d'assemblage. Cela est par exemple fait automatiquement avec les étiquettes de programme, comme dans l'exemple suivant:

posit: EQU $

Les arguments apparaissant dans ASM peuvent être reliés par différents opérandes arithmétiques, comme par exemple:

MVI A,12+2*3

Les opérandes arithmétiques sont:

+X	nombre positif
-X	nombre négatif, correspond à 0-X (toujours 16 bits!)
X+Y	addition de valeurs 16 bits
X-Y	soustraction de valeurs 16 bits
X*Y	produit de X*Y
X/Y	division des arguments
X MOD Y	fonction reste de la division X/Y

Peuvent en outre être utilisées ces deux instructions de décalage:

X SHL Y	décale la valeur 16 bits X de Y emplacements vers la gauche. Les bits expulsés sont perdus.
X SHR Y	décale la valeur 16 bits X de Y emplacements vers la droite. Les bits expulsés sont perdus.

Il y a encore les opérateurs logiques:

NOT X	négation logique de l'argument X
X AND Y	liaison logique ET des arguments X et Y
X OR Y	liaison logique OU des arguments X et Y
X XOR Y	liaison logique OU EXCLUSIF des arguments X et Y

Toutes ces possibilités vous paraissent peut-être superflues au premier abord. Mais elles ne le sont pas, tout au contraire: elles sont même très utiles. Si vous voulez par exemple charger dans l'accumulateur l'octet fort d'une étiquette d'adresse, cela est très simple à réaliser en décalant logiquement cette étiquette de 8 bits vers la droite:

```
MVI A,Label SHR 8
```

Si vous voulez avoir uniquement les 8 bits inférieurs, vous devez expressément masquer les 8 bits supérieurs car sinon l'accumulateur, registre 8 bits, recevrait une valeur 16 bits. Ce masquage se présente ainsi:

```
MVI A,Label AND OFFH
```

Nous avons choisi cette application comme exemple car elle est certainement une de celles qui se présenteront le plus souvent. Mais bien sûr, des milliers d'autres applications sont imaginables.

Il y a encore les pseudo-codes d'opération

ORG, EQU, END, DS, DB, DW, SET, IF et ENDIF.

(Les pseudo-codes d'opération sont des instructions assembleur qui sont lues et traitées par l'assembleur pendant l'opération d'assemblage. Le nom "pseudo-codes d'opération" vient du fait que ces instructions apparaissent comme les codes d'opération normaux dans le programme assembleur mais qu'elles ne sont pas assemblées et qu'elles ne peuvent pas être comprises par l'unité centrale.)

Nous n'expliquerons pas en détail les différents pseudo-codes d'opération mais nous en évoquerons brièvement l'effet et le mode

de fonctionnement. Cela sort en effet du cadre d'un ouvrage sur CP/M.

ORG <adresse de départ>

Cette instruction définit l'adresse de départ du programme à assembler. ORG devrait toujours être la première instruction d'un programme.

EQU <valeur>

EQU affecte une valeur fixe à une étiquette. La <valeur> peut être également une expression arithmétique.

Exemple: Diskon: EQU Diskr+O55H

DS

Cette instruction réserve dans le programme une zone libre pour y placer par exemple des données. Le Program Counter est augmenté en conséquence.

Exemple: Diskm: DS 100

Dans cet exemple, 100 octets seront gardés disponibles. L'adresse de départ est Diskm, l'adresse de fin Diskm+99. La prochaine instruction commence en Diskm+100.

DB

Ici, différents octets à définir sont placés dans la mémoire. Les différents octets doivent être séparés entre eux par une virgule. Des chaînes de caractères sont également possible.

Exemples: Disktxt: DB "Introduire disquette"
Diskt2: DB 23,32,0,3,122

DW

DW définit des mots, c'est-à-dire des valeurs 16 bits. Ici aussi, différentes valeurs 16 bits peuvent être séparées entre elles par une virgule.

Exemple: Diskadr: DW O3AB9H,label1

END

END marque la fin du programme à assembler. Pour des besoins spécifiques, il est possible d'indiquer ici une adresse de départ pour le traitement du programme assemblé.

SET

L'instruction SET permet comme l'instruction EQU d'affecter une valeur à une étiquette. Avec une différence cependant: alors qu'avec l'instruction EQU l'affectation est fixe, les étiquettes définies avec l'instruction SET peuvent encore être modifiées lors de l'assemblage. Les étiquettes définies avec SET ne sont pas non

plus affichées dans la première colonne lors de l'assemblage.

Exemple: Label1: SET Alpha
Label1: SET Label1 + 32

Mais avec tout cela nous n'avons pas encore épuisé les possibilités de ASM. Comme beaucoup d'autres assembleurs puissants, ASM offre aussi la possibilité d'assembler de façon conditionnelle, c'est-à-dire de ne faire décoder et convertir en langage machine une section déterminée du programme assembleur que sous certaines conditions. Cela est obtenu avec les instructions IF et ENDIF. (L'expérience révèle cependant que ces instructions sont rarement utilisées.)

Supposons que vous vouliez écrire un programme pour lire deux valeurs et les additionner ou les multiplier entre elles. L'algorithme est identique dans les deux cas. Vous pouvez maintenant écrire un programme qui interrogera un flag lors de l'assemblage pour savoir si le programme doit additionner ou multiplier.

```
;
Multi  EQU 0 ;le programme doit additionner
ORG 0100H
;
;lecture des valeurs
CALL lecture
;
IF Multi
  CALL Multir ;appeler la routine de multiplication
ENDIF
IF NOT Multi
  CALL Addir ;appeler la routine d'addition
ENDIF
CALL Sortie
END
```

Si vous voulez maintenant que la routine multiplie, il vous suffit de remplacer dans la première ligne la valeur pour Multi par NOT 0.

Après avoir traité aussi en détail de l'assembleur, revenons encore brièvement sur l'appel de ASM:

Comme vous vous le rappelez, vous ne devez pas indiquer l'extension ".ASM" lorsque vous appelez ASM. Les deux fichiers <Nom de fichier>.PRN et <Nom de fichier>.HEX sont produits sur la disquette. Cela peut cependant également être empêché! On peut également indiquer à partir de quel lecteur de disquette doit être chargé le fichier à assembler ou sur quel fichier doivent être placés les deux fichiers à produire. La syntaxe de l'instruction ASM est:

ASM <Fichier>.<Fichier ASM><Fichier HEX><Fichier PRN>

A part le nom de fichier <Fichier>, toutes les indications sont facultatives mais si elles sont fournies, elles doivent l'être dans l'ordre ci-dessus. Les options suivantes sont possibles:

Pour Fichier ASM: A ou B.
A ou B indiquent le lecteur de disquette du fichier source.

Pour Fichier HEX: A, B ou Z.
A et B sont ici également l'indication du lecteur de disquette sur lequel le fichier portant le nom <Nom de fichier>.HEX a été placé. Il est également possible d'indiquer à l'assembleur de ne pas produire de fichier HEX. C'est intéressant par exemple lorsque vous voulez simplement faire un parcours-test pour faire tester la syntaxe, par exemple. Entrez alors en seconde position l'option 'Z'.

Pour fichier PRN: A, B, X ou Z.

A et B sont à nouveau l'indication du lecteur de disquette sur lequel le fichier <Nom de fichier>.PRN doit être produit. Ce protocole peut cependant également être interdit en entrant 'Z' (voir plus haut). Vous pouvez en outre faire sortir le fichier de protocole sur l'écran avec l'option 'X'.

Supposons que vous vouliez charger le fichier source appelé Asterix à partir du lecteur de disquette B, afficher le fichier de protocole à l'écran, ne pas conserver le fichier HEX, par exemple pour que l'assembleur vérifie la syntaxe de votre programme. Vous entrerez alors:

```
ASM Asterix.BZX
```

Vous voyez, c'est très simple. Mais concluons maintenant le chapitre assembleur; vous disposez certainement maintenant de suffisamment d'informations sur l'assembleur ASM.

VII.7 Le travail avec SUBMIT et XSUB

Il arrive souvent que l'on doive toujours et encore entrer les mêmes séquences d'instructions. Imaginez par exemple que vous vouliez assembler un fichier source déterminé, convertir ensuite le fichier HEX en un fichier COM puis lancer enfin ce nouveau fichier COM pour le tester.

C'est justement dans des cas semblables qu'il est conseillé de créer un fichier SUBMIT qui contiendra toutes les instructions nécessaires et les exécutera automatiquement les unes à la suite des autres. SUBMIT ne signifie pas autre chose que transmettre: le programme SUBMIT transmet cette chaîne d'instructions au CCP, le Console Command Processor qui est chargé de l'entrée au clavier et de l'exécution des instructions entrées. Cela se passe ainsi:

Lorsque vous dites à SUBMIT quel fichier doit être transmis comme fichier SUBMIT, SUBMIT crée un fichier provisoire $$$.SUB. Dans ce fichier provisoire figure tout ce que contient le fichier avec quelques compléments. Avant que CCP n'accepte une instruction venant du clavier, on regarde d'abord s'il n'y pas un fichier temporaire $$$.SUB sur le lecteur de disquette annoncé. Si c'est le cas, une ligne d'instructions est chargée dans le buffer du CCP et cette ligne est effacée du fichier. Cela est répété jusqu'à ce que le fichier $$$.SUB soit vide et ce n'est qu'après qu'une instruction sera à nouveau entrée au clavier.

Comme vous le savez, l'instruction DIR ne vous indique malheureusement pas combien de place il reste encore sur la disquette. Cette information peut être obtenue avec l'instruction STAT. Nous allons créer un fichier SUBMIT qui réunisse les instructions DIR et STAT de façon à ce que nous puissions apprendre avec une seule instruction quel est le contenu de la disquette et quelle est sa capacité encore disponible. Nous créerons pour cela avec un traitement de texte, par exemple avec ED, ce fichier SUBMIT et nous l'appellerons DISK.SUB. Tous les fichiers SUBMIT doivent absolument porter la marque ".SUB".

Bien, notre premier fichier SUBMIT se présente donc ainsi:

```
STAT
DIR
```

Nous reconnaissons qu'il s'agit là d'un fichier SUBMIT très court mais c'est en forgeant qu'on devient forgeron! Lançons donc ce premier fichier SUBMIT avec:

```
SUBMIT disk
```

Vous voyez que nous ne devons pas indiquer l'extension ".SUB".

Notre lecteur de disquette a l'air très occupé mais nous savons bien pourquoi. (Création du fichier $$$.SUB, copie du contenu de DISK.SUB, suppression ligne par ligne des instructions du fichier $$$.SUB etc...) Et ça marche.

Rappelons-nous ce à quoi nous voulions arriver. Nous voulions assembler un fichier, en faire un fichier COM grâce à l'instruction LOAD et enfin le lancer. Cette séquence d'instructions se présenterait à peu près ainsi:

ASM <Nom de fichier>.<Options>
LOAD <Nom de fichier>
<Nom de fichier>

Les termes <Nom de fichier> et <Options> figurent ici en fait comme des variables. Mais comment peut-on remplir ces variables? Les développeurs du programme SUBMIT ont pris en compte ce problème et ont eu une bonne idée pour rendre les fichiers SUBMIT plus faciles à organiser. Vous pouvez transmettre ces données lors de l'appel du fichier SUBMIT en les séparant entre elles par au moins un espace. Ces données peuvent être de simples nombres ou même des termes plus compliqués tels que des noms de fichier. Les données sont numérotées en commençant par 1. Dans le fichier SUBMIT, ces variables doivent être marquées par un signe dollar $ suivi directement du numéro de la variable. Certaines variables peuvent être répétées indéfiniment dans le fichier SUBMIT. Lors de la création du fichier SUBMIT, les noms de variables seront alors automatiquement remplacés par les données, de sorte que CCP pourra les traiter sans aucun problème. Notre fichier SUBMIT se présente maintenant ainsi:

ASM $1,$2
LOAD $1
$1 $3

Vous voyez que la première entrée, c'est-à-dire la première donnée apparaît trois fois dans le fichier SUBMIT. Si nous lançons par exemple ce fichier SUBMIT avec notre programme Dump et que nous ne voulions pas de fichier de protocole (un fichier HEX doit être créé car LOAD en a besoin), l'appel pourrait se présenter ainsi si nous avions sauvegardé le fichier SUBMIT sous le nom de "ASSEM.SUB":

SUBMIT ASSEM DUMP AAZ DUMP.COM

La dernière indication indique quel fichier doit être dumpé, dans notre exemple DUMP.COM. Si vous essayez et que cela marche, vous aurez fait le premier pas vers l'utilisation des fichiers SUBMIT Le premier pas, car SUBMIT offre encore d'autres possibilités!

Vous vous êtes peut-être déjà demandé: que faire si un signe dollar doit apparaître dans le fichier SUBMIT? Très simple: il faut faire ce qu'on fait habituellement en informatique dans un cas semblable: on écrit deux signes dollar à la suite. Supposons par exemple que vous vouliez supprimer tous les fichiers temporaires. Vous y parviendrez au clavier avec:

ERA *.$$$

Dans un fichier SUBMIT cela a l'air un peu plus terrible:

ERA *.$$$$$$

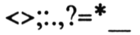
mais cela signifie la même chose. Comme le signe dollar apparaît normalement assez rarement, le travail supplémentaire occasionné n'est pas très important.

Lorsqu'une ligne d'instruction est copiée du fichier $$$.SUB dans la mémoire du CCP, cette ligne est également représentée d'abord à l'écran. Le clavier est ensuite interrogé pour savoir si une touche a été actionnée. Si c'est le cas ou si l'instruction dans la mémoire d'instruction n'est pas exécutée ou n'est pas exécutée correctement, SUBMIT interrompt le déroulement de l'instruction et supprime le fichier $$$.SUB.

Lorsqu'une ligne du fichier SUBMIT commence par un caractère qui ne doit normalement pas apparaître dans un nom de fichier, cette ligne est bien affichée à l'écran mais elle n'est pas exécutée. Il vous est facile de faire afficher de cette manière sur l'écran des commentaires ou des instructions destinées à l'utilisateur. Il serait par exemple possible d'ajouter ce qui suit dans notre fichier SUBMIT ASSEM.SUB:

```
:le fichier est maintenant en cours d'assemblage.
:veuillez patienter un peu!!
ASM $1,$2
LOAD $1
$1 $3
```

Et l'exécution de ce fichier SUBMIT se fera déjà de façon plus sympathique. Les caractères qui ont cet effet sont:


<>;:.,?=*_

Avec SUBMIT les variables se limitent cependant au CCP, c'est-à-dire qu'on ne peut pas entrer de données pour un programme appelé tel que PIP ou ED. Cela serait cependant parfois également très intéressant, par exemple pour copier certains fichiers au moyen de PIP ou pour donner certaines instructions à ED. Ce "défaut" peut lui aussi être éliminé. Il existe pour cela le programme XSUB qui intercepte toutes les entrées faites au clavier et qui se charge de l'interrogation du fichier $$$.SUB. Lorsque nous voulons par

exemple donner des indications à PIP (ce qui fonctionne bien sûr également de manière directe), cela pourrait se présenter ainsi:

```
XSUB
PIP
b:=A:*.COM
```

Dans la dernière ligne vous devez appuyer une fois sur ENTER pour quitter la routine transitoire PIP. Avec PIP cette application n'est cependant pas encore si intéressante que cela car on aurait pu écrire directement l'instruction autrement, par exemple: PIP b:=a:*.COM pour copier tous les fichiers COM du lecteur de disquette A sur le lecteur de disquette B. Cette application devient cependant très intéressante avec ED. Vous voyez qu'il suffit d'appeler XSUB une fois dans la première ligne. Les indications qui sont normalement données au clavier viendront maintenant du fichier $$$.SUB.

Les anciennes versions CP/M 2.2 recélaient encore une erreur dans le programme SUBMIT. Lorsqu'on transférait des données au moyen de XSUB, on ne pouvait pas transmettre de caractères de commande bien que cela fût prévu dans le manuel. C'est très ennuyeux car avec PIP et ED par exemple, on a souvent à envoyer le caractère de commande ^Z. Mais réjouissez-vous, cette erreur a été éliminée dans votre version CP/M 2.2.

Nous espérons que vous avez suffisamment compris l'application et la manipulation des fichiers SUBMIT pour être maintenant en mesure de créer vos propres fichiers SUBMIT lorsque vous en ressentirez le besoin.

Nous avons maintenant traité de l'assembleur et des fichiers SUBMIT. Nous allons encore évoquer brièvement un désassembleur qui figure également sur la disquette. Vous appelez ce désassembleur avec DDT. Si vous voulez par exemple faire imprimer le désassemblage du fichier d'instructions FILECOPY.COM, entrez simplement:

DDT filecopy.com

Avant d'appuyer sur ENTER, vous devez naturellement donner avec <CTRL>/P l'instruction de tout sortir sur l'imprimante.

Désassembler signifie le contraire d'assembler: les codes machine sont convertis en codes mnémoniques intelligibles pour un humain.

VII.8 L'organisation de la mémoire

Nous avons déjà évoqué plusieurs fois des termes tels que BDOS, CCP, BIOS, etc... La plupart d'entre vous ont certainement déjà compris qu'il s'agit de concepts très importants. Certains lecteurs se sont certainement déjà démandé où tout cela figure dans la mémoire de l'ordinateur. Répétons donc ce que signifient ces termes:

CCP

CCP = Console Command Processor. Cette zone règle l'entrée au clavier et elle contient les instructions résidentes, soit: DIR, ERA, REN, SAVE, TYPE et USER.

BDOS

BDOS = Basic Disk Output System. Cette partie règle l'échange de données entre ordinateur et lecteurs de disquette.

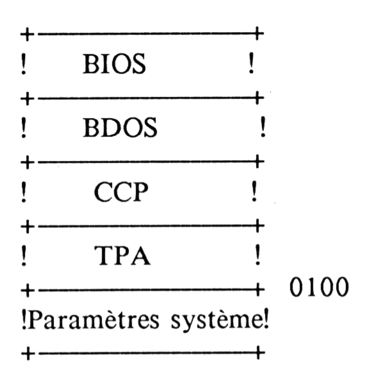
BIOS

BIOS = Basic Input/Output System. BIOS et BDOS travaillent en étroite collaboration et c'est pourquoi on trouve aussi souvent les deux partenaires sous un nom collectif: FDOS.

TPA

La zone pour les programmes utilisateur dans la mémoire de l'ordinateur est appelée TPA. Les programmes à exécuter et les

données qui doivent être traitées avec ces programmes sont gérés dans cette mémoire.



Voilà comment se présente en gros l'organisation de la mémoire sous CP/M. La zone utilisateur (TPA) commence à l'adresse &0100.

Comme l'explication de la programmation sous CP/M sortirait du cadre de cet ouvrage, nous évoquerons simplement brièvement la programmation sous CP/M à l'aide des routines BDOS et BIOS.

Le BDOS contient une collection de sous-programmes qui rendent possible le travail de base avec des fichiers sur disquette et avec certains périphériques. Comme CP/M n'est pas tout jeune, il existe par exemple encore des routines pour l'entrée et la sortie sur carte perforée. Cela est certainement encore très utilisé mais sur notre CPC nous pouvons ignorer délibérément de telles routines.

Pour appeler une routine BDOS, on doit suivre les conventions standard suivantes:

Dans le registre C du processeur est transmis au BDOS le numéro de code de la sous-routine voulue. Dans le registre E ou dans le double registre DE pour les valeurs 16 bits, des données sont transmises à la sous-routine, si nécessaire. Lorsque les registres ont été alimentés comme il convient, on saute à l'adresse 5 à

travers une instruction de sous-programme (CALL). BDOS détermine alors d'après le registre C l'adresse à laquelle se trouve la routine voulue. Vous voyez qu'il est ainsi possible de rendre des programmes exécutables sur vraiment tous les ordinateurs, car on respecte des règles fixes. Toutes les choses qui varient d'un ordinateur à l'autre, comme l'entrée/sortie au clavier et à l'écran sont réglées par des routines du BDOS spécialement développées pour chaque ordinateur. Lors du développement d'un nouvel ordinateur, il "suffit" donc d'écrire ces routines pour que tous les programmes CP/M tournent sur cet ordinateur.

Deux conditions doivent cependant être remplies. L'ordinateur doit posséder un processeur 8080 ou compatible avec le 8080 et il faut qu'il soit possible de lire les programmes existants. Si cette dernière condition n'était pas remplie, les programmes CP/M pourraient toujours tourner en théorie sur cet ordinateur. D'ailleurs l'expérience révèle que le transfert de programmes existants ne constitue pas le problème le plus important. Cela peut se faire a) par programmation du controller ou b) en transmettant au moyen d'un programme relativement simple les programmes existants à travers une interface commune aux deux ordinateurs, par exemple à travers une RS 232. Mais même lorsqu'il n'existe pas d'interface commune, les bidouilleurs trouvent toujours un moyen: le port joystick peut par exemple être utilisé comme port d'entrée pour la transmission de programmes.

Mais revenons au BDOS. Si le BDOS doit renvoyer des valeurs de la routine BDOS au programme principal, les résultats 8 bits se trouveront dans l'accumulateur du processeur, alors que les valeurs 16 bits se trouveront dans le double registre HL. Pour les valeurs 16 bits, le bit inférieur se trouvera en outre dans l'accumulateur alors que le bit supérieur sera dans le registre B.

Si vous voulez par exemple sortir un caractère sur la console, on utilise pour cela la routine BDOS No 2. Nous sortons maintenant un point d'interrogation (en mnémoniques Z80):

```
LD C,2  ;sortie au clavier
```

```
LD E,"?";charger point d'interrogation
CALL 5 ;appeler BDOS
```

Si vous voulez vraiment programmer en langage machine sous CP/M, il existe un littérature technique spéciale pour les programmeurs sous CP/M. Nous vous donnons ici une liste des routines BDOS avec les paramètres d'entrée et de sortie pour que vous puissiez faire des expériences. Pour une programmation sérieuse on a certainement besoin de connaissances plus complètes sur les différentes routines.

Nom de la routine	C	Reçoit	Fournit
Déclencher démarrage chaud	0		
Entrée console	1		A:entrée
Sortie console	2	E:caractère	
Lire barres perforées	3		A:caractère
Perforer barres	4	E:caractère	
Caractère à l'imprimante	5	E:caractère	
Entrée directe console	6	E:255	A:0
et sortie		E:caractère	A:caractère
Interroger IOBYTE	7		A:IOBYTE
Fixer IOBYTE	8	E:IOBYTE	
Sortir chaîne	9	DE:=>chaîne de caractères	
Entrée du buffer	10	DE:=>buffer	A:caractère
Etat console	11		A:=0 (pas de touche)
Version CP/M	12		HL:Version
Reset système disquette	13		
Fixer lecteur de référence	14	E:numéro LW	
Ouverture fichier	15	DE:infos	A:255 (erreur)
Fermeture fichier	16	DE:infos	A:255 (erreur)
Chercher première entrée	17	DE:infos	A:255 (erreur)
		sinon:	A:caractère
Entrée suivante	18		A:255/caractère
Supprimer fichier	19	DE:infos	A:255 (erreur)
Lire information	20	DE:infos	A:0 (OK)
Ecrire information	21	DE:infos	A:0 (OK)

Créer fichier	22	DE:infos	A:255 (erreur)
Changer nom fichier	23	DE:infos	A:255 (erreur)
Déterminer lecteur actif	24		HL:vecteur lect.
Déterminer lecteur de réf.	25		A:No lecteur
Fixer buffer de données	26	DE:buffer	
Déterminer table affectat.	27		HL:adresse
Protéger lecteur de référ.	28		
Lecteurs protégés	29		HL:vecteur lect.
Fixer options fichier	30	DE:infos	A:255 (erreur)
Déterminer paramètres disk	31		HL:adresse
Gérer No utilisateur	32	E:255 E:numéro	A:numéro
Lire information	33	DE:infos	A:0(OK)
Ecrire information	34	DE:infos	A:0(OK)
Déterminer taille fichier	35	DE:infos	(dans buffer)
Fixer descripteur	36	DE:infos	(dans buffer)
Restaurer lecteur	37	DE:vectr lect.	A:0

Cette table doit simplement montrer comment peut se présenter à peu près la programmation avec BDOS et quelles possibilités vous offre BDOS. Comme nous l'avons déjà indiqué, cette table ne suffit pas à vous permettre de programmer car des connaissances sur le fonctionnement interne des différentes routines sont également nécessaires.

Si BDOS contient des routines telles que "protéger lecteur de référence", les tâches de BIOS sont cependant encore plus spécialisées. BIOS se compose comme BDOS d'une liste de routines mais celles-ci ne servent qu'à l'entrée et à la sortie de données. On trouve dans le BDOS certaines routines qui remplissent les mêmes fonctions que des routines du BIOS, par exemple pour déterminer l'état de la console.

BIOS contribue de façon décisive au succès de CP/M car il permet aux programmes CP/M de faire exécuter au moyen de routines spéciales les entrées et sorties sur la console alors que celles-ci varient considérablement en fonction de l'ordinateur.

BIOS est subdivisé grossièrement en quatre zones:

- Interface avec le BDOS et les programmes CP/M (par exemple pour les fichiers SUBMIT)
- Entrée et sortie à travers les périphériques appelés par BDOS
- Stockage provisoire dans des buffers de données qui seront ensuite fournis à la demande (fichiers SUBMIT)

Comme vous le savez, BDOS et BIOS peuvent être décalés dans la mémoire. Les routines du BDOS sont appelées par la transmission du numéro de routine dans le registre C et par l'appel de l'adresse mémoire fixe &0005. L'appel de routines BIOS se présente un peu différemment: il y a une table de saut dont l'ordre est fixé d'avance mais dont l'adresse de début peut être décalée. Mais examinons d'abord cette table de saut:

```
00+décalage: JMP BOOT     ;démarrage à froid
03+décalage: JMP WBOOT    ;démarrage à chaud
06+décalage: JMP CONST    ;interroge état console
09+décalage: JMP CONIN    ;entrée console
0C+décalage: JMP CONOUT   ;sortie console
0F+décalage: JMP LIST     ;sortie sur imprimante
12+décalage: JMP PUNCH    ;écrire barres perforées
15+décalage: JMP READER   ;lire barres perforées
18+décalage: JMP HOME     ;tête sur piste 0
1B+décalage: JMP SELDSK   ;sélection du lecteur
1E+décalage: JMP SETTRK   ;fixer piste
21+décalage: JMP SETSEC   ;sélectionner section d'informations
24+décalage: JMP SETDMA   ;sélection du buffer de données
27+décalage: JMP READ     ;lire section d'informations
2A+décalage: JMP WRITE    ;écrire section d'informations
2D+décalage: JMP LISTS    ;interroge état imprimante
30+décalage: JMP SECTRAN  ;traduire numéro d'informations
```

Le décalage correspond au décalage dans la mémoire sur lequel nous

reviendrons plus loin. Si une des fonctions indiquées n'est pas utilisée -comme c'est certainement le cas sur le CPC pour les deux routines qui concernent les cartes perforées-, les adresses mémoire correspondantes peuvent être simplement remplies par

RET ! NOP ! NOP

Pour que la mémoire ne soit pas décalée après cette routine. Un appel de cette routine entraînerait donc immédiatement un abandon de la routine avec RET; il ne se passera donc rien. Pour déterminer l'adresse de départ (le décalage) de BIOS, il faut savoir que l'adresse 0 de la page de base du système contient un saut au vecteur de démarrage à chaud (le deuxième vecteur de la table). Cela permet donc de déterminer facilement l'adresse de départ comme nous allons essayer de le montrer maintenant. Lancez d'abord CP/M puis chargez le désassembleur:

DDT

Pour désassembler la zone de mémoire commençant à l'adresse 0, entrons dans l'ordinateur ce qui suit:

I0000

et nous obtenons rapidement la réponse de DDT:

```
0000  JMP  AD03
0003  ADD  C
0004  NOP
0005  JMP  8F00
0008  JMP  B982
```

etc... Ce qui nous intéresse, c'est donc la première ligne: on saute à l'adresse &AD03 et la table de saut du BIOS commence donc à l'adresse &A300. C'est tout ce qui nous intéresse pour nos programmes. Si nous voulons sauter à un vecteur, nous calculons l'adresse d'entrée avec:

Adresse:=numéro de saut * 3 + &A300

Avec les routines BDOS, vous pouvez, exactement comme pour les routines BIOS, transmettre ou recevoir des paramètres. Si des valeurs doivent être transmises aux routines BIOS, les valeurs 8 bits sont transmises dans le registre C, les valeurs 16 bits sont attendues dans le registre BC. Vous voyez ici déjà une différence avec les routines BDOS qui attendent les paramètres dans le registre E ou dans le registre double DE. Les paramètres que les routines BIOS renvoient au programme principal sont, comme pour les routines BDOS, fournies dans l'accumulateur pour les valeurs 8 bits et dans le double registre HL pour les valeurs 16 bits. Les esprits aiguisés ont certainement déjà trouvé une possibilité très intéressante: comme les différentes routines BIOS sont appelées à travers des vecteurs situés dans la RAM, on peut bien sûr modifier ces vecteurs, c'est-à-dire les détourner vers une routine propre de l'utilisateur qui le cas échéant peut également sauter au programme d'origine.

Nous allons nous intéresser encore aux autres particularités de la page de base (adresses &0000 à &00FF) du système. Avec l'instruction I0, nous avons fait désassembler la zone &0000 à &0016. Un certain nombre d'autres instructions JMP apparaissent alors: particulièrement importante est l'instruction de saut à l'adresse 5; vous devriez normalement comprendre maintenant à quoi nous faisons allusion car c'est vers l'adresse 5 que nous avons dirigé notre appel de sous-programme lorsque nous avons voulu appeler une routine BDOS! BDOS doit donc se situer sur le CPC à l'adresse &8F00 puisque c'est à cette adresse que nous renvoie l'instruction de saut à l'adresse &0005. Examinons donc de plus près cette zone de la mémoire:

I8F00

Pour que ce ne soit pas trop simple, nous voyons qu'il nous faut sauter encore une fois, à l'adresse &95A2. Ne nous laissons pas démonter et effectuons encore ce saut:

I95A2

Pour être bref: on saute encore à l'adresse &9F06 puis à l'adresse &9F11. Ce n'est qu'après que commence progressivement le décodage du registre C pour déterminer la routine à laquelle il faut sauter. Cela nous conduirait certainement trop loin si nous voulions suivre plus longtemps ce mécanisme. Vous pouvez aisément le faire vous-même avec le DDT et une longue veille dans votre chambre.

VII.9 L'instruction SETUP

L'instruction SETUP n'existe que sous CP/M 2.2; elle peut cependant également intéresser les possesseurs d'un CPC 6128 car CP/M 2.2 tourne aussi sur le 6128.

Vous n'appréciez peut-être pas la couleur de fond standard que sélectionne CP/M? Ou peut-être aimeriez-vous que le CPC ne se présente pas avec ce message impersonnel:

CP/M 2.2 - Amstrad Consumer Electronics plc.

Dans ce cas, l'instruction SETUP est faite pour vous. SETUP permet d'adapter le CP/M CPC à vos besoins personnels. SETUP est commandé par un menu et il interroge les différents points par un dialogue à l'écran. Vous pouvez ainsi obtenir que CP/M soit beaucoup plus gentil à votre égard et vous salue par un amical:

Salut Alex, vieille branche - tu as encore besoin de moi?

Mais appelons maintenant l'instruction SETUP. Nous parcourrons ensuite plus en détail les différents points du menu.

Après que vous ayez entré SETUP, apparaît sur l'écran:

SETUP V2.0

** Initial command buffer:empty

Is this correct (Y/N):_

La question Is this correct (Y/N) vous est posée pour chaque élément de l'instruction SETUP. Si vous entrez Y pour oui, les paramètres affichés seront acceptés alors que vous pouvez encore les corriger en entrant N pour non.

Vous voyez que notre "Initial command buffer" est vide (empty). Lorsque vous écrivez quelque chose dans le Initial command buffer, ce buffer sera exécuté lors du démarrage de CP/M. Si vous voulez par exemple avoir sur l'écran, immédiatement après le chargement de CP/M, un catalogue de la disquette système, pour avoir un aperçu des instructions transitoires, vous pouvez réaliser cela à travers ce Initial command buffer. Vous répondez donc "Non" à la question "Is this correct" pour pouvoir adapter votre Initial command buffer. Après que vous ayez actionné la touche N, apparaît sur l'écran:

Enter new initial command buffer:_

Nous allons remplir notre buffer avec l'instruction DIR. Pour qu'une ligne puisse être exécutée, celle-ci doit bien sûr être

terminée par une retour de chariot; mais comment entrer un retour de chariot? Pour tous les caractères ASCII dont le code est inférieur à 32, vous devez utiliser les séquences de contrôle ^A à ^Z ainsi que ^(, ^/, ^), ^> et ^-. Pour entrer par exemple le retour de chariot, vous devez utiliser le caractère ^M. (le retour de chariot a le code ASCII 13 et la lettre M est la treizième de l'alphabet!) Entrez donc ce qui suit dans le Initial command buffer:

DIR^M

Après que vous ayez appuyé sur la touche ENTER apparaît à nouveau la question Is this correct (Y/N):_, à laquelle vous pouvez maintenant répondre par un Y pour oui.

EnsuiteestafficheéetinterrogéleSIGN-ON-STRING.SIGN-ON-STRING contient à nouveau un bon nombre de caractères de commande qui effectuent la fixation du mode 80 caractères ainsi que des couleurs du fond et du bord. Ici aussi, les caractères ASCII dont les codes sont inférieurs à 32 doivent être entrés avec ^<code>. Les caractères ASCII correspondants pour: fixer la couleur du fond, fixer le mode 80 colonnes etc... figurent dans le manuel de l'utilisateur, dans la table des caractères ASCII. Vous pouvez créer ici votre propre message d'interrogation, comme par exemple notre "Salut Alex...".

Après que vous ayez défini le Initial command buffer de même que le SIGN-ON-STRING, vous pouvez affecter au clavier des caractères alternatifs ou supplémentaires. C'est à cela que sert le point

Keyboard translations

Les indications que vous devez fournir dans ce point du menu correspondent exactement à l'instruction BASIC KEY DEF. Les tables de codes correspondantes ainsi que les codes du clavier figurent

également dans le manuel de votre ordinateur. Le point suivant du menu, Keyboard expansions, a également les mêmes effets que l'instruction BASIC KEY.

A la suite de cela vous sont montrées les affectations standard des périphériques d'entrée/sortie que vous pouvez également éditer de la même façon. Suivent quelques données techniques sur le lecteur de disquette (temps d'arrêt total du moteur, etc.), ainsi que des données concernant les canaux d'entrée/sortie du processeur. Après que vous ayez accepté au clavier toutes ces données, vous obtenez la question:

Do you want to update your system disc (Y/N):_

Si vous répondez à cette question par Y, toutes les données que vous avez fournies seront sauvées sur disquette (si vous appuyez sur la touche "N", les données seront perdues). La dernière question que vous pose CP/M est:

Do you want to restard CP/M (Y/N):_

Cette possibilité vous permet de tester directement à l'écran les effets des modifications que vous avez apportées. Répondez pour cela "Y" pour oui. CP/M est alors à nouveau chargé et si vous avez par exemple modifié le SIGN-ON-STRING ou le Initial command buffer, cela sera immédiatement exécuté.

Vous comprenez certainement maintenant l'intérêt de CP/M. Nous espérons donc que cet ouvrage vous a beaucoup aidé et vous aidera encore à découvrir et utiliser CP/M.

VIII. Correction de PIP

VIII.1 Description des erreurs

Il y a dans PIP un petit détail qui peut donner des sueurs froides. Si l'on travaille avec PIP pour copier des données et que l'on change de disquette sans faire de démarrage à chaud avec CTRL-C, on obtient le message d'erreur "BDOS ERROR R/O". Cela est très ennuyeux car il faut alors appuyer sur la touche CTRL-C et tout entrer à nouveau.

Comme nous n'aimons pas nous énerver et que nous ne voulions pas non plus que vous ayez à vous énerver à l'avenir, nous allons vous proposer une solution en reléguant le message "BDOS ERROR R/O" au magasin des accessoires. Cela est certainement très intéressant pour vous, surtout parce que cela permettra de voir directement comment on utilise DDT et comment on peut examiner et modifier des fichiers COM CP/M.

VIII.2 La correction des erreurs

Le problème évoqué plus haut sera résolu en corrigeant un peu le programme PIP.COM. Le mieux est que vous n'effectuiez d'abord cette correction que sur votre copie de sécurité. Imaginez-vous en effet ce qui se passerait si vous faisiez une erreur et que vous n'ayez pas de copie de CP/M!

Nous allons maintenant faire appeler au début la fonction CP/M disk reset, de façon à ce qu'une nouvelle disquette qui aurait été éventuellement placée dans le lecteur puisse être initialisée. Il nous faut pour cela fouiner dans le fichier PIP.COM, de préférence avec DDT.

Entrez:

DDT PIP.COM

Le programme commence, comme tout autre fichier COM, à l'adresse &0100. Vous pouvez faire désassembler cette zone:

```
-I0100
 0100 JMP 04CE
 0103 RET
```

Le début du programme PIP se trouve donc en 04CE. Mais nous voulons cependant encore ajouter un disk reset. C'est pourquoi nous allons ajouter cette petite routine à la fin du programme PIP et nous la ferons appeler au début de PIP. La fin du programme PIP se trouve en &1DB1. Entrons maintenant cette petite routine:

```
-a1db2
1DB2  MVI C,0D
1DB4  CALL 0005
1DB7  JMP 04CE
1DBA  (appuyez sur la touche ENTER)
```

Cette routine appelle donc la routine disk reset. Il nous faut maintenant encore indiquer au début de PIP qu'il faut d'abord sauter à cette routine:

```
-a0100
0100  JMP 1DB2
0103  (appuyez sur la touche ENTER)
```

VIII.3 Sauvegarde du nouveau PIP

Le programme PIP est maintenant corrigé. Vous devriez maintenant l'essayer (de préférence avec toutes les options possibles). S'il ne fonctionne pas conformément à la description, c'est que quelque chose n'a pas marché. Mais vous avez (heureusement!) encore une copie de sécurité de CP/M que vous allez pouvoir d'abord copier avant de faire une seconde tentative. (Vous voyez combien il est important de disposer dans un placard d'une copie de sécurité. Ce n'est pas seulement lors de manipulations qu'une disquette peut vous quitter pour toujours. Imaginez par exemple que votre petite fille ou votre petit garçon décident de faire joujou avec!)

Nous devons maintenant sauvegarder à nouveau notre programme sur disquette car il ne figure pour le moment corrigé que dans la mémoire. Pour cela, vous devez d'abord quitter DDT en appuyant sur la touche CTRL-C.

Le programme peut alors être à nouveau sauvegardé au moyen de l'instruction SAVE. Cela est réalisé avec l'instruction suivante:

```
SAVE 29 PIP.COM
```

Si la disquette n'est pas protégée contre l'écriture, tout est terminé et PIP a été corrigé. L'erreur ne se reproduira plus.

Vous voyez que même des programmes aussi importants et complets que le système d'exploitation CP/M, faits par des sociétés aussi importantes que Digital Research peuvent encore présenter des erreurs. Il est tout simplement impossible d'éliminer toute erreur d'un programme.

IX. Toutes les instructions CP/M

IX.1 ASM (CP/M 2.2)

ASM.COM

Constitue à partir d'un fichier source contenant des instructions 8080 ou Z80 un fichier objet en format HEX

Format d'entrée

ASM FICHIER
ASM FICHIER.BBZ

Description

Ce programme constitue un fichier HEX à partir d'un fichier contenant des instructions 8080 ou Z80. Ce fichier, appelé fichier objet, peut être ensuite chargé dans le système avec le programme LOAD, comme instruction transitoire.

Application

Le programme ASM cherche toujours un fichier avec la marque ASM de sorte que cette marque ne doit pas être indiquée. Trois paramètres peuvent être entrés avec l'instruction: shp. S est mis pour source, ce qui désigne le lecteur de disquette (A..P) dans lequel figure le fichier source. H est mis pour HEX et il correspond à l'emplacement du nom du lecteur de disquette sur lequel doit être créé le fichier HEX. Si vous voulez interdire la création du fichier HEX, entrez un Z. P est mis pour PRN et vous pouvez indiquer ici le lecteur de disquette sur lequel le fichier PRN doit être créé. Pour interdire ce fichier, entrez à nouveau Z ou un X si vous voulez voir le fichier PRN à l'écran.

IX.2 COPYSYS

COPYSYS.COM

Copie les pistes système et CP/M3.SYS sur une disquette.

Format d'entrée:

COPYSYS (CP/M 3.0)

Description:

Pour commencer avec une disquette CP/M 3.0, les deux parties de CP/M doivent être présentes: la première partie sur les pistes système et la deuxième partie comme fichier CPM3.SYS. Les disquettes qui ne sont pas utilisées pour le chargement de CP/M n'ont besoin d'aucune des deux parties.

Application:

Entrez COPYSYS puis ENTER. On vous demande à partir de quel lecteur de disquette doit se faire la copie:

Source drive name (or return for default)

Entrez la lettre du lecteur de disquette dans lequel figure votre disquette avec le système d'exploitation CP/M (lecteur de disquette A: ou 1, normalement). Si la disquette est déjà dans le lecteur de disquette, entrez simplement ENTER. Entrez ensuite le nom du lecteur de disquette de destination et appuyez sur la touche ENTER. Si vous n'avez pas encore placé la disquette dans le lecteur, on vous y invite. Une fois que les pistes système ont été écrites, vous pouvez encore copier CPM3.SYS. On vous demande:

Do you wish to copy CPM3.SYS?

Répondez "Y" et le fichier CP/M sera copié.

Remarque:

COPYSYS est l'instruction standard pour copier les pistes système sous CP/M 3.0. Ce fichier COM ne figure malheureusement sur aucune des disquettes CPM fournies avec la machine. Le paquet de programmes DISCKIT offre cependant la possibilité de formater des disquettes dans n'importe quel format et de copier le système.

DISCKIT3 est un paquet de programmes très pratique pour l'utilisateur que la firme AMSTRAD a développé spécialement pour le CPC 6128. Malheureusement, au moment de l'écriture du présent ouvrage, n'existait encore aucune documentation sur ce paquet de programmes. Il est certain en tout cas que vous pouvez avec DISCKIT3 formater des disquettes sans problème et copier le système CP/M 3.0. Nous vous invitons pour plus d'informations sur DISCKIT3 à vous reporter à son manuel d'utilisation.

IX.3 DATE (CP/M 3.0)

DATE.COM

Indique la date et l'heure et permet leur entrée

Format d'entrée

DATE
DATE CONTINUOUS
DATE SET
DATE MM/DD/YY HH:MM:SS

Description

Ce programme vous permet d'entrer une date et une heure dans votre système ou d'appeler ces deux informations.

Application:

Si vous entrez simplement l'instruction DATE, vous obtenez l'affichage de la date et de l'heure

Tue 05/06/85 23:49:17

Il s'agit d'un affichage fixe. Si vous voulez comparer l'heure avec celle d'une autre horloge, entrez l'instruction DATE avec l'option "C". Vous voyez alors un affichage permanent de l'heure mais vous ne pouvez pas faire tourner en même temps un autre programme ni entrer quelques instructions que ce soit. Pour interrompre le programme, appuyez sur une touche quelconque.

Vous pouvez entrer la date et l'heure de deux façons: avec l'option SET; le programme vous demande alors les valeurs les unes après les autres et vous pouvez sauter une entrée avec ENTER. L'autre possibilité se présente ainsi:

DATE 05/06/85 23:49:17

Entrez une heure qui avance d'une ou deux minutes sur l'heure qu'il est vraiment. Lorsqu'arrivera l'heure que vous avez entrée, vous n'aurez plus qu'à appuyer sur la touche espace pour que l'heure tourne.

Si votre ordinateur ne dispose pas d'horloge sur batterie comme c'est le cas pour l'AMSTRAD CPC 6128, il vous faut renouveler cette entrée après chaque démarrage du système. Vous pouvez vous faciliter ce travail en écrivant l'instruction DATE SET dans le fichier PROFILE.SUB.

IX.4 DDT

DDT.COM

Programme de débuggage pour charger, exécuter, modifier et tester des programmes

Format d'entrée:

DDT

DDT Fichier.COM

Description:

Le programme DDT remplace le CCP et charge alors le fichier indiqué dans la mémoire de travail. DDT permet d'interroger l'adresse suivant l'adresse finale d'un programme (NEXT) et le contenu du compteur de programme (Program Counter = PC).-

Application:

Le programme doit figurer sur la disquette en tant que DDT.COM. Si DDT est entré, le programme attend encore qu'on entre un nom de fichier. En entrant Dn, on peut faire sortir sur l'écran la zone de mémoire commençant à l'adresse n. Pour mettre fin au programme, on utilise de préférence l'instruction GO.

IX.5 DEVICE (CP/M 3.0)

DEVICE.COM

Indique et modifie les voies de communication avec les périphériques

Format d'entrée:

DEVICE NAMES
DEVICE VALUES
DEVICE LST:=XXXXXX[NOXON,1200]
DEVICE CONOUT:=XXX,XXX
DEVICE CON:[PAGES]
DEVICE CON:[COLUMNS=nn, LINES=mm]

Description:

Ce programme est utilisé pour afficher et modifier les canaux d'entrée/sortie du système d'exploitation. Les trois canaux logiques sont: CON:, LST: et AUX:. On peut également diriger un canal logique sur plusieurs périphériques. Les données venant de l'ordinateur peuvent par exemple être envoyées simultanément sur l'écran et sur l'imprimante. DEVICE vous permet aussi de déterminer le nombre de lignes et de colonnes sur l'écran.

Application:

Les noms des unités de sortie sont fixés par les constructeurs. Si vous entrez DEVICE avec l'option NAMES, vous obtiendrez une liste des noms et des vitesses de transmission des différents périphériques. Si vous entrez DEVICE VALUES, vous obtenez un affichage se présentant à peu près ainsi:

Current assignments:

CONIN:= CRT
CONOUT:= CRT
AUXIN:= LPT

AUXOUT:= LPT
LST:= CEN

CONIN: et CONOUT: signifient CONsol INput, entrée clavier et CONsole OUTput, sortie clavier. Si CON: figure seul, cela se réfère aux deux. LST: signifie LIST DEVICE et désigne l'imprimante.

L'option PAGE vous permet de déterminer combien de lignes et de colonnes sont actuellement affichées à l'écran. Si vous entrez une certaine vitesse de transmission sous la forme [nnn], les échanges de données avec le périphérique correspondant s'effectueront désormais à cette vitesse. Avec l'instruction DEVICE vous obtenez l'affichage de DEVICE NAMES et VALUES en même temps.

Notez que l'instruction DEVICE est remplacée sous CP/M 2.0 par l'instruction STAT DEV: (on devrait plutôt dire que l'instruction STAT a été divisée sous CP/M 2.2 parce qu'elle était devenue trop complexe.)

IX.6 DIR (CP/M 2.2 et CP/M 3.0)

DIR.COM et instruction intégrée (CP/M 3.0)

Montre différents noms de fichiers et le catalogue

Format d'entrée:

Instruction intégrée:

DIR
DIR A:
DIR FICHIER.XXX
DIR AB* *

Instruction transitoire:

DIR[OPTION]
DIR FICHIER.XXX[OPTION]
DIR AB*.*[OPTION]

Description:

DIR est un programme très pratique pour retrouver des fichiers sur une disquette ou surtout sur un disque dur. (Malheureusement, encore aucun fabricant ne propose de disque dur pour un ordinateur AMSTRAD) La version intégrée travaille avec chaque lecteur de disquette et chaque zone utilisateur. Les signes "Joker" "*" et "?" peuvent être utilisés. Si aucune option n'est entrée, tous les fichiers non-système de la zone utilisateur correspondante sont affichés.

DIR comme instruction transitoire est encore beaucoup plus puissant: affichage des fichiers avec tampon dateur, tri dans l'ordre alphabétique, etc... Voici maintenant une énumération des différentes options qui peuvent être entrées entre crochets. Si des fichiers système figurent également sur la disquette, apparaît le message:

SYSTEM FILE(S) EXIST

OPTIONS:

Option	Fonction
ATT	sort les attributs de fichier définis par l'utilisateur
DATE	affiche les fichiers avec la date et l'heure
DIR	affiche les fichiers non-système
DRIVE=ALL	affiche les fichiers de tous les lecteurs de disquette
DRIVE=A	affiche les fichiers sur le lecteur de disquette A:
DRIVE=(A,B)	affiche les fichiers sur les lecteurs indiqués
EXCLUDE	affiche tous les fichiers, sauf celui indiqué
FF	décalage de page avant sortie du catalogue
FULL	affiche les fichiers avec description détaillée
LENGTH=n	titre après n lignes
MESSAGE	affiche en permanence les lecteurs et les zones USER
NOPAGE	affiche contenu du catalogue sans arrêt en fin de page
NOSORT	affiche les fichiers non triés
RO	affiche les fichiers Read Only
RW	affiche les fichiers Read Write
SIZE	indiquer également la taille de chaque fichier
SYS	n'affiche que les fichiers système
USER=ALL	affiche les fichiers de toutes les zones USER
USER=5	affiche les fichiers de la zone USER
USER=(3,5)	affiche les fichiers des zones USER

IX.7 DIRSYS (CP/M 3.0)

Instruction intégrée

Affiche les fichiers qui ont été déclarés comme fichiers système

Format d'entrée:

DIRSYS
DIRSYS B:
DIRSYS FICHIER.XXX
DIRSYS AB?*.*

Description:

DIRSYS affiche tous les fichiers système. Cela peut bien sûr être également obtenu avec DIR et ses options mais DIRSYS est plus rapide puisque c'est une instruction intégrée. Vous pouvez abréger cette instruction en:

DIRS

IX.8 DUMP (CP/M 2.2 et CP/M 3.0)

DUMP.COM

Affiche les fichiers non-texte en hexadécimal et en ASCII

Format d'entrée:

DUMP
DUMP FICHIER.XXX

Description:

DUMP est surtout intéressant pour les programmeurs qui travaillent en assembleur. Les fichiers de texte peuvent être affichés à l'écran ou sur l'imprimante avec TYPE, les fichiers du type COM, REL ou OVR avec DUMP.

Vous trouverez au chapitre VII. une description détaillée des fichiers-COM ASM, DDT et DUMP.

IX.9 ERA(SE) (CP/M 2.2 et CP/M 3.0)

Instruction intégrée et ERASE.COM

Supprime un, plusieurs ou tous les fichiers de la disquette

Format d'entrée:

Instruction intégrée:

ERASE
ERASE FICHIER.XXX
ERASE AB?*.*

Instruction transitoire:

ERASE FICHIER.XXX[CONFIRM]

Description:

Avec l'instruction ERASE, le fichier indiqué est supprimé s'il n'est pas "Read Only" ou protégé contre l'écriture. Vous ne pouvez effectuer de suppression qu'à l'intérieur de la zone USER. Si vous utilisez "*" ou "?", l'instruction repète l'ordre de suppression et vous demande de confirmer la suppression.

Application:

Sans entrer d'options, ERASE peut travailler à partir de n'importe quel lecteur de disquette. Pour les entrées avec options, le programme ERASE.COM doit figurer sur le lecteur de disquette. ERASE peut être abrégé en ERA. L'option "CONFIRM" peut être abrégée en "C".

IX.10 FORMAT (CP/M 2.2 et CP/M 3.0)

FORMAT.COM

Formate une disquette

Format d'entrée:

FORMAT

Description:

Toute disquette que vous utilisez sur votre ordinateur doit être correctement formatée. Le programme détruit lors du formatage toutes les données pouvant figurer sur une disquette. FORMAT n'est pas un programme CP/M standard. C'est pourquoi il est possible qu'il ne figure pas sur votre disquette système. Recherchez alors le programme COPY.COM et voyez s'il vous permet de formater une disquette.

Application:

FORMAT travaille jusqu'à ce que vous l'interrompiez avec Control C (^C). Vous pouvez ainsi formater plusieurs disquettes l'une à la suite de l'autre.

Remarque:

Avec le CPC 6128, on ne vous propose pas, sous CP/M 3.0, de FORMAT comme fichier COM. FORMAT est intégré comme sous-programme dans le paquet d'utilitaires DISCKIT3. Si vous voulez malgré tout avoir FORMAT sous la forme d'un fichier COM, par exemple pour l'intégrer dans des fichiers SUBMIT, vous pouvez toujours copier le fichier COM de la disquette CP/M 2.2.

IX.11 GENCOM (CP/M 3.0)

GENCOM.COM

Crée une version spéciale de CP/M 3.0

Format d'entrée:

GENCOM

Description:

Est utilisé par les programmeurs pour adapter une version de CP/M ou pour intégrer des parties de programme supplémentaires dans CP/M.

IX.12 GET (CP/M 3.0)

GET.COM

Retire des données d'un fichier au lieu de les retirer du clavier

Format d'entrée:

GET CONSOLE INPUT FROM FILE FICHIER.XXX
GET CONSOLE INPUT FROM CONSOLE
GET CONSOLE INPUT FROM FILE FICHIER.XXX[SYSTEM]
GET FILE FICHIER.XXX[NOECHO]

Description:

L'instruction GET vous permet, au lieu d'entrer les données ou instructions, de les faire venir d'un fichier quelconque.

Application:

La première ligne ci-dessus ordonne à CP/M de traiter la prochaine instruction ou le prochain programme qui sera entré au clavier. Dès qu'une entrée est nécessaire, CP/M ira la chercher dans le fichier FICHIER.XXX. Si le programme est terminé ou s'il ne reste plus d'instruction dans le fichier, CP/M revient à l'entrée au clavier.

L'instruction de la seconde ligne ordonne à CP/M d'attendre à nouveau les instructions du clavier. Cette instruction peut figurer dans le fichier FICHIER.XXX ou être entrée par l'utilisateur.

La troisième forme de l'instruction détourne immédiatement l'entrée au clavier vers le fichier indiqué pour que celui-ci puisse lire et traiter les instructions au clavier. Si vous entrez comme option (NOECHO), les instructions en train d'être exécutées ne seront pas affichées à l'écran.

IX.13 HELP (CP/M 3.0)

HELP.COM et HELP.HLP

Donne des explications sur les différentes instructions CP/M, avec des exemples

Format d'entrée:

HELP
HELP terme
HELP terme sous-concept
HELP terme[option]
HELP .sous-concept
HELP [option]

Description:

Le programme HELP.COM vous permet de recevoir de l'aide et des informations sur les instructions et programmes CP/M. Vous ne pouvez pas appeler HELP si vous êtes en train de travailler avec un autre programme.

Application:

Lorsque vous appelez HELP, une série de termes vous est proposée. Vous pouvez y rechercher ce qui vous intéresse et entrer ce terme. Vous pouvez abréger le terme à ces deux premières lettres. Si l'information fournie compte plus de 23 lignes, l'affichage s'arrête dès que l'écran est plein et vous passez à la page-écran suivante en appuyant sur la touche espace. Si vous entrez un Control P (^P) avant d'appeler la fonction HELP, toutes les informations seront imprimées sur l'imprimante.

Si vous voulez modifier le texte figurant dans le fichier HELP.HLP, vous devez d'abord entrer HELP avec l'option EXTRACT puis vous pouvez écrire votre texte et vous créez enfin le nouveau fichier, lisible par CP/M, avec l'option CREATE.

L'instruction est particulièrement utile pour les débutants car on n'est plus ainsi obligé d'avoir toujours sous la main le manuel de CP/M. Cet exemple donné par Digital Research a été suivi dans d'autres paquets de logiciels connus; combien de programmes ont maintenant ce qu'on appelle des écrans HELP et qui sont censés vous faciliter le travail avec ces paquets de programmes.

Surtout si vous voulez apprendre à travailler sous CP/M 3.0, vous devriez étudier de plus près ces textes HELP. Ils vous apprendront à utiliser CP/M 3.0 et vous feront connaître la plupart des instructions disponibles.

IX.14 HEXCOM (CP/M 3.0)

HEXCOM.COM

Crée un fichier exécutable directement, avec la marque COM

Format d'entrée:

HEXCOM FICHIER
HEXCOM

Description:

Vous avez besoin de ce programme lorsque vous programmez en assembleur. HEXCOM convertit les fichiers créés avec'le programme MAC.COM sous le format INTEL HEX en fichiers exécutables avec la marque COM.

IX.15 INITDIR (CP/M 3.0)

INITDIR.COM

Prépare un catalogue de disquette à être doté du tampon avec la date et l'heure

Format d'entrée:

INITDIR B:

Description:

CP/M 3.0 permet de doter les fichiers d'un tampon indiquant la date et l'heure. Les informations correspondantes sont stockées dans une section spéciale du catalogue. C'est pourquoi le catalogue de chaque disquette doit être préparé en conséquence en appelant le programme INITDIR. Le programme SET vous permet de choisir le mode de marquage des fichiers.

Application:

Le mieux est d'entrer INITDIR pour des disquettes neuves car les entrées du catalogue ont besoin de place. Si vous utilisez INITDIR avec une disquette contenant déjà des fichiers, faites-en d'abord une copie de sécurité. Si en effet le programme est interrompu en plein travail, par exemple par une panne de courant, vous perdrez toutes les entrées qui figuraient sur le catalogue.

Vous voyez que ce n'est pas que pour la disquette système qu'il est recommandé de réaliser une copie de sécurité.

IX.16 LIB (CP/M 3.0)

LIB.COM

Crée et modifie une "bibliothèque" de sous-programmes

Format d'entrée:

LIB FICHIER.XXX[options]

Description:

De nombreux compilateurs et les assembleurs RMAC MACRO 80 utilisent la méthode des sous-programmes séparés, le plus souvent dotés de la marque REL ou IRL. LIB vous permet de réunir les principaux sous-programmes dans une "bibliothèque".

IX.17 LINK (CP/M 3.0)

LINK.COM

Crée un fichier exécutable à partir de modules de sous-programmes

Format d'entrée:

LINK FICHIER1, FICHIER2, FICHIER3,..[options]

Description:

De nombreux compilateurs et les assembleurs RMAC MACRO 80 utilisent la méthode des sous-programmes séparés, le plus souvent dotés de la marque REL ou IRL. LINK vous permet de réunir ces sous-programmes en un fichier COM.

IX.18 LOAD (CP/M 2.2 et CP/M 3.0)

LOAD.COM

Crée un fichier COM à partir d'un fichier HEX

Format d'entrée:

LOAD FICHIER (sans marque de fichier!)

Description:

L'assembleur ASM crée sous forme d'un fichier hexadécimal le code réalisé en deux passages. La forme de ce fichier hexadécimal a été définie par la société INTEL. Le programme LOAD traite maintenant ce fichier de façon à réaliser à partir des données hexadécimales indiquées en code ASCII un fichier COM exécutable. Il faut faire attention à ne pas indiquer le type de fichier ".HEX". L'assembleur sauve en effet obligatoirement le fichier HEX sous le nom Fichier.HEX.

Application:

LOAD doit être normalement appelé après chaque opération d'assemblage car on ne peut rien tirer du seul code hexadécimal d'un programme. Pour l'application précise de cette instruction et des instructions du même type, vous pouvez vous reporter au chapitre VII.

IX.19 MAC (CP/M 3.0)

MAC.COM

Macro-assembleur pour des programmes écrits en assembleur

Format d'entrée:

MAC FICHIER
MAC FICHIER $options

Description:

Trois fichiers et une bibliothèque de macros sont créés par le macro-assembleur MAC. Le programme source porte la marque ASM et la "bibliothèque" la marque LIB. Le code de programme assemblé figure dans le fichier portant la marque HEX, la table des symboles est dans le fichier SYM et le listing à imprimer dans le fichier PRN. Au lieu des parenthèses habituelles, c'est par le signe dollar ($) que sont marquées les options.

IX.20 MOVCPM (CP/M 2.2)

MOVCPM.COM

Décalage du système par unités de 256 octets

Format d'entrée:

MOVCPM <facteur>*

Description:

L'instruction MOVCPM sert à décaler (à reloger) le système CP/M 2.2 par pas de 256 octets. Le <facteur> doit être compris entre 64 et 179. Si vous entrez 179 comme <facteur>, c'est un CP/M standard qui sera produit. De même, si vous entrez MOVCPM 178*, c'est un CP/M 2.2 décalé de 256 octets vers le bas qui sera produit. Le <facteur> indique en unités de 256 octets la place mémoire disponible pour les instructions CP/M (transitoires) et les programmes utilisateur.

Application:

S'il s'avère nécessaire, pour une raison quelconque, de décaler le système, par exemple parce que BIOS/BDOS se chevauche avec le programme utilisateur, cela peut être obtenu avec MOVCPM. Le système décalé peut alors être sauvegardé avec l'instruction SYSGEN (voir IX.33).

IX.21 PATCH (CP/M 3.0)

PATCH.COM

Installe des modifications dans CP/M 3.0 ou dans des programmes

Format d'entrée:

PATCH FICHIER
PATCH FICHIER n

Application:

Le programme PATCH vous permet d'apporter des modifications au système CP/M et de les intégrer dans CP/M ou dans un des programmes transitoires. Ce procédé s'appelle "patching". Avec PATCH, les modifications sont automatiquement intégrées dans le programme correspondant. Lorsqu'il y a plusieurs patchs, on se réfère au patch correspondant en utilisant des numéros de référence. Les numéros de référence commencent à 1.

IX.22 PIP (CP/M 2.2 et CP/M 3.0)

PIP.COM

Copie des fichiers et transfère des fichiers entre périphériques

Format d'entrée:

PIP
PIP fichier objet=fichier source[option]
PIP B:=Fichier.XXX
PIP Tout.txt=TEXTE1.TXT,TEXTE2.TXT...
PIP AB1?*.*=AB2?*.*[option]

Description:

PIP est certainement le programme le plus puissant de CP/M. Il vous permet de copier des fichiers et ce d'une disquette dans une autre mais aussi d'une zone utilisateur à une autre. Il vous permet aussi de créer un fichier unique à partir de plusieurs fichiers, de sauvegarder automatiquement les fichiers traités dernièrement, de tout convertir en majuscules et de faire annuler le huitième bit. Si vous souhaitez des informations plus précises, veuillez vous reporter au chapitre VI.

Application:

PIP peut être employé avec ou sans options. Lorsque PIP est prêt à exécuter des instructions, il affiche son signe "prêt", l'étoile (*). Il crée un fichier objet exactement identique au fichier source, sauf si des options spéciales ont été choisies. Les crochets doivent figurer directement après le nom de fichier, sans espace. Si vous ne connaissez pas certaines lettres du nom de fichier ou si vous voulez copier plusieurs fichiers d'une même catégorie, vous pouvez utiliser également le caractère "?" à la place d'un caractère indéterminé du nom ou le caractère "*" pour tous les caractères suivants. Voici quelques exemples:

Votre lecteur de disquette standard est A:

PIP B:=FICHIER.XXX[V]

Vous copierez ainsi le fichier FICHIER.XXX de A: dans B: et PIP vérifiera le nouveau fichier.

PIP B:=FICHIER?.*[V]

Vous copierez ainsi une série de fichiers portant le nom FICHIER, quelle que soit leur marque de fichier, de A: dans B:.

PIP B:=*.*[V]

Tous les fichiers d'un lecteur de disquette seront ainsi transférés sur un autre et vérifiés.

PIP NEWFILE.XXX=OLDFILE.XXX

Cette instruction copiera le fichier OLDFILE.XXX dans le fichier nouvellement créé NEWFILE.XXX sur le même lecteur de disquette. Les deux fichiers figureront ensuite en même temps sur cette disquette.

PIP TOUT;TXT=TEXTE1.TXT,TEXTE2.TXT,TEXTE3.TXT,...

Vous regroupez ainsi plusieurs fichiers dans un seul fichier.

PIP B:[G] =TEXTE.TXT

Le fichier TEXTE.TXT est copié sur le lecteur de disquette B: dans la zone utilisateur "5".

Options

A Archiv Ne copie que les fichiers qui ont été créés ou modifiés depuis le dernier archivage. La fonction tampon de date et heure doit avoir été fixée.

C ConfirmCP/M demande pour chaque fichier si ce fichier peut être copié.

Dn Delete Suppression de n colonnes. PIP supprime tous les caractères qui figurent dans le fichier après la n-ième colonne. N'utiliser que pour des fichiers de texte.

E Echo Affiche texte sur l'écran. Le contenu des fichiers en train d'être copiés est affiché sur l'écran. Ne pas utiliser en même temps que le paramètre "N". N'utiliser que pour les fichiers de texte.

F Interdire le caractère pour "page suivante" (form feed). Quelques imprimantes ont besoin d'un (^L) pour avancer à la page suivante. Si votre imprimante n'en a pas besoin, vous pouvez empêcher les caractères de commande correspondants avec "F".

Gn Get Amener un fichier de ou vers la zone utilisateur "n". Cette indication doit figurer directement à la suite du premier nom de fichier et c'est la seule qui puisse figurer à cet emplacement. Exemple:

PIP B:[G5]=TEXTE.TXT[G1]

Copiera le fichier TESTE.TXT de la zone utilisateur "1" vers la zone utilisateur "5".

H Hex Transfert de données hexadécimales. Vous devriez toujours entrer cette option lorsque vous voulez transférer des fichiers HEX. PIP teste alors le contenu pour voir s'il est écrit dans un format INTEL correct.

I Ignore Ignorer le caractère fin de fichier (EOF) dans les fichiers HEX. Entrez cette option pour chaque fichier, sauf pour le dernier. L'option "H" doit être fixée en même temps que l'option "I". Utilisez pour le dernier fichier le paramètre "H":

PIP FICHIER.HEX=PROG1.HEX[I],PROG2.HEX[H]

K Interdire l'affichage du nom de fichier lors de la

copie.

L	Convertir toutes les majuscules en minuscules. N'utilisez cette option que pour les fichiers de texte et utilisez en plus le paramètre "Z" pour les fichiers WordStar.
N	Numérotage continu des lignes d'un fichier. Les numéros commencent à 1 pour la première ligne et sont augmentés de 1 par ligne. Les nombres ne peuvent comporter plus de 6 chiffres, les chiffres non-utilisés apparaissent comme espaces. Le nombre est suivi d'un double point et d'un espace. Ne pas utiliser en même temps que les options "E" ou "N". Pour les fichiers WordStar, entrer en plus l'option "Z".
N2	Numérote les lignes pour un programme BASIC.
O Object	Transfert de fichiers objet. Est utilisé pour transférer les fichiers non-texte et non-COM. Ne pas utiliser pour les fichiers de texte.
Pn Page	Fixe la longueur de page. La valeur pré-fixée est de 60 lignes par page. Vous devriez entrer en même temps l'option "F" pour que les caractères pour fin de page (^L) soient supprimés. N'utiliser que pour les fichiers de texte.
Qxxxx^Z	Fin de la copie après cette chaîne de caractères. PIP copie un fichier jusqu'à la chaîne de caractères entrée inclusivement. Utilisez l'instruction avec deux lignes d'instructions pour que la chaîne de caractères ne soit pas convertie en majuscules:

```
PIP
CON:=TEXTE.TXT[Astuce]
```

Si vous écrivez tout dans la première ligne à la suite de PIP, l'entrée apparaîtra en majuscules.

R Copie les fichiers système. Ces fichiers ne sont pas affichés par DIR et ne sont normalement pas copiés par PIP. "R" permet de lever cette limitation.

Sxxxx^Z Commencer à cette chaîne de caractères. PIP commence la copie à cette chaîne de caractères. Ecrivez cette chaîne de caractères sur deux lignes, sinon la chaîne de caractères sera convertie en majuscules. N'utiliser que pour les fichiers de texte.

Tn Tab Fixer l'espacement du tabulateur. CP/M travaille normalement avec un pas de huit caractères. Cela ne marche cependant pas lors de l'impression d'un fichier et le pas doit être défini. "n" indique le nombre d'espaces pour un pas de tabulation. N'utiliser que pour les fichiers de texte.

U Conversion en majuscules. Convertit toutes les minuscules en majuscules. N'utiliser que pour les fichiers de texte. Pour les fichiers WordStar, utiliser en plus l'option "Z".

V Verify Ce paramètre fait que PIP comparera précisément le fichier copié avec le fichier source.

W Write Effacer même les fichiers comportant l'attribut (RO). Ces fichiers ne peuvent normalement qu'être lus mais non modifiés ou supprimés. Avec l'option "W", ces fichiers seront supprimés sans demande de confirmation.

Z Zero Supprimer le bit de parité. Le jeu de caractères ASCII n'utilise que sept bits. Le huitième bit est utilisé par différents programmes pour des tâches différentes. La fonction "Z" n'est pas nécessaire lors de la copie vers des unités ASCII telles que CON: ou LST:.

IX.23 PUT (CP/M 3.0)

PUT.COM

Ecrit dans un fichier les données pour l'imprimante ou pour l'écran

Format d'entrée:

PUT CONSOLE OUTPUT TO FILE FICHIER.XXX[OPTION]
PUT PRINTER OUTPUT TO FILE FICHIER.XXX[OPTION]
PUT CONSOLE OUTPUT TO CONSOLE
PUT PRINTER OUTPUT TO PRINTER

Description:

CP/M envoie normalement les données pour l'écran à l'écran et les données pour l'imprimante à l'imprimante. Avec l'instruction PUT les données peuvent être en plus écrites dans un fichier.

Application:

Les deux premières lignes ci-dessus ordonnent à CP/M d'ouvrir un fichier portant le nom FICHIER.XXX et d'y écrire tout ce qui est destiné à l'écran ou à l'imprimante. Une fois l'opération terminée, le programme revient au mode CP/M normal. Les deux dernières lignes interrompent le programme PUT. Vous pouvez les utiliser par exemple lorsque vous avez un fichier SUBMIT qui utilise l'instruction PUT et qui doit ensuite retourner au mode normal.

Avec l'option NO ECHO vous interdisez l'affichage des données transférées à l'écran. Avec l'option FILTER vous convertissez tout caractère de contrôle en un caractère imprimable.

Avec l'option SYSTEM, non seulement les données mais aussi la ligne d'instructions iront dans le fichier.

IX.24 REN(AME) (CP/M 2.2 et CP/M 3.0)

RENAME instruction intégrée ou RENAME.COM

Changer le nom d'un fichier

Format d'entrée:

RENAME
RENAME FICHIER2.XXX=FICHIER1.XXX
RENAME AB?*2=AB?*1

Description:

RENAME vous permet de modifier le nom -et seulement le nom- d'un fichier. Le contenu n'est pas affecté.

Application:

Si le fichier ne se trouve pas sur le lecteur de disquette standard, le nom du lecteur de disquette peut être également entré. Si vous entrez RENAME sans options, le programme vous les réclamera.

IX.25 RMAC (CP/M 3.0)

RMAC.COM

Macro-assembleur pour la programmation en assembleur

Format d'entrée:

RMAC FICHIER
RMAC FICHIER $OPTIONS

Description:

Trois fichiers plus une bibliothèque de macros sont créés par le macro-assembleur RMAC. Le programme source a la marque ASM et la "bibliothèque" a la marque LIB. Le code programme assemblé figure dans le fichier portant la marque REL, la table de symboles dans le fichier SYM et le listing imprimable dans le fichier PRN. Avec le programme LINK peut être créé un programme exécutable à partir du fichier REL.

Application:

Au lieu d'être marquées par les parenthèses habituelles, les options sont marquées par le signe dollar. Un espace doit figurer avant le signe dollar.

IX.26 SAVE (CP/M 2.2 et CP/M 3.0)

SAVE.COM

Sauvegarde le contenu de la mémoire de travail dans un fichier

Format d'entrée:

SAVE

Description:

L'instruction SAVE crée un fichier avec le contenu d'une zone mémoire déterminée. Avant la sauvegarde, un autre programme doit avoir écrit quelque chose dans la mémoire.

Application:

Vous devez appeler SAVE AVANT de charger un autre programme dans la mémoire de travail. SAVE se place automatiquement à la fin de CP/M et redonne ensuite le contrôle à CP/M. Vous pouvez alors faire tourner le programme. Dès qu'il a fini, SAVE reprend alors automatiquement le contrôle. On vous demande le nom du fichier et les adresses de début et de fin dans la mémoire.

IX.27 SET (CP/M 3.0)

SET.COM

Fixe les attributs de fichier, permet l'entrée d'un mot de code, donne un nom aux disquettes et sélectionne le mode des tampons de date et d'heure

Format d'entrée:

SET FICHIER.XXX[option]
SET AB?*.*[option]
SET B:[option]
SET[option]

Description:

Le programme SET est employé pour diverses tâches. Parmi les plus importantes: la fixation du mode archiv et la possibilité de déclarer des fichiers et des lecteurs de disquette entiers comme fichiers ou lecteurs de disquette Read Only.

Un nom propre peut être entré pour chaque disquette ainsi qu'en outre un mot de protection pour chaque disquette et pour chaque fichier. Les noms de disquette peuvent être affichés avec SHOW.

Une troisième possibilité est de doter les fichiers de tampons pour la date et l'heure.

Lisez le chapitre V si vous souhaitez de plus amples informations.

IX.28 SETDEF (CP/M 3.0)

SETDEF.COM

Montre ou définit la piste de recherche et active ou désactive l'affichage à l'écran

Format d'entrée:

SETDEF
SETDEF B:
SETDEF[option]

Description:

Vous lancez normalement un programme transitoire en en tapant le nom puis en appuyant sur ENTER. Si vous voulez appeler le programme sur un autre lecteur de disquette, entrez également le nom du lecteur de disquette.

Si vous travaillez par exemple toujours sur le lecteur de disquette B: mais que vos fichiers CP/M sont sur le lecteur de disquette A:, vous pouvez indiquer à CP/M où il doit rechercher les fichiers. Cela s'appelle prescrire une piste de recherche. Vous pouvez ainsi indiquer à CP/M de chercher d'abord sur le lecteur de disquette A:, ensuite sur le lecteur de disquette C: et enfin seulement sur le lecteur de disquette standard.

IX.29 SHOW (CP/M 3.0)

SHOW.COM

Affiche les options d'une disquette

Format d'entrée:

SHOW[option]
SHOW B:[option]

Description:

SHOW vous permet d'afficher ce qu'on appelle les "données techniques" d'une disquette. Vous pouvez recevoir les informations suivantes: place libre sur une disquette, nombre d'entrées libres dans le catalogue, le nombre des zones utilisateur et le numéro de la zone utilisateur sélectionnée. Vous pouvez obtenir également des indications sur la capacité globale de la disquette, la taille d'un bloc, la taille d'un secteur et le nombre de secteurs par piste. Le nom d'une disquette peut également être affiché.

Lorsque vous utilisez CP/M 2.2, tenez compte du fait que cette instruction est également réalisée au moyen de STAT.

IX.30 SID (CP/M 3.0)

SID.COM

Un programme de débuggage pour charger un programme assembleur, le modifier et le tester

Format d'entrée:

SID
SID FICHIER.XXX
SID FICHIER.XXX TEXTE.SYM

Description:

SID vous permet de charger les fichiers COM ou HEX dans la mémoire de travail, de les y examiner et modifier. Lorsque vous chargez en même temps un fichier SYM, vous pouvez faire tourner le programme tout en le contrôlant précisément. SID peut également traduire les programmes pouvant fonctionner en mnémoniques INTEL 8080.

IX.31 SUBMIT (CP/M 2.2 et CP/M 3.0)

SUBMIT.COM

Permet l'entrée d'instructions à partir d'un fichier, au lieu du clavier

Format d'entrée:

SUBMIT
SUBMIT FICHIER.SUB
SUBMIT FICHIER FICHIER1 FICHIER2 FICHIER3...

Description:

Les instructions que vous entrez normalement au clavier peuvent également être écrites dans un fichier SUBMIT pour être ensuite traitées automatiquement. Si plus aucune instruction ne figure dans le fichier, le contrôle revient à CP/M et vous pouvez continuer comme avant.

CP/M cherche lors de chaque nouveau démarrage un fichier portant le nom PROFILE.SUB et il exécute les instructions figurant dans ce fichier s'il l'a trouvé.

Application:

Vous écrivez simplement un fichier SUBMIT avec votre programme de texte, en écrivant chaque instruction sur une ligne. Ce fichier doit porter la marque SUB, sinon il ne sera pas reconnu par SUBMIT.

IX.32 STAT (CP/M 2.2)

STAT.COM

Fournit dans une liste la place mémoire occupée ainsi que les états protection et système des fichiers indiqués. STAT fournit en outre dans une liste l'état de tous les lecteurs de disquette actuellement actifs sur le système.

Format d'entrée:

STAT
STAT FICHIER.XXX
STAT FICHI??.*
STAT DEV:
STAT DSK:
STAT USR:
STAT FICHIER.XXX $SYS

Application:

STAT vous permet en fait de faire sortir les options de vos disquettes et des fichiers qui y figurent. L'instruction est semblable à l'instruction CP/M 3.0 SHOW. Vous pouvez faire sortir la place mémoire libre sur une disquette, la longueur de n'importe quels fichiers en enregistrements ou par exemple les zones utilisateur utilisées. L'instruction STAT contient donc également la fonction de l'instruction CP/M 3.0 DEVICE. Avec STAT DSK:, vous pouvez examiner toutes les caractéristiques de n'importe quelle disquette. Vous pouvez en outre fixer avec l'instruction STAT des fichiers sur $R/O (Read Only), $R/W (Read and Write), $SYS (fichier système) et $DIR (fichier catalogue). Ces états peuvent également être affichés au moyen de l'instruction STAT. STAT contient donc également l'instruction CP/M 3.0 SET. Comme vous le voyez, STAT est très souple et d'utilisation très large.

STAT en détail:

STAT sort dans une liste l'état de tous les lecteurs de disquette

actuellement activés dans le système:

<lecteur>: <protection>, Space: <n>k

STAT USR: sort le numéro actuel d'utilisateur ainsi que toutes les zones utilisateur occupées sur la disquette.

Active User: 5
Active Files: 0 1 3 5 7

STAT DEV: correspond à l'instruction CP/M 3.0 DEVICE et sort une liste des périphériques d'entrée/sortie. Une réaffectation est possible.

CON: is CRT:
RDR: is TTY:
PUN: is TTY:
LST: is LPT:

STAT <canal>=<périphérique> affecte à un canal un nouveau périphérique d'entrée/sortie.

STAT CON:=CRT:, LST:=TTY:

STAT <lecteur> $R/O ou STAT <lecteur> $R/W fixe le lecteur indiqué sur l'état lire seulement ou sur l'état écriture/lecture.

STAT <lecteur>: sort la place mémoire actuellement disponible sur la disquette indiquée.

Bytes remaining on A: 5k

STAT <nom de fichier.XXX>, éventuellement avec des jokers, sort la place occupée et l'état protection ou système du ou des fichiers indiqués.

Recs Bytes EXT Acc
 144 18k 2 R/W A:(D2.BAS)
  14 2k 1 R/W A:DISC.BAS

Si un fichier est entre parenthèses, c'est qu'il s'agit d'un fichier système, donc d'un fichier qui a été doté de l'option $SYS. Il n'apparaît pas dans le catalogue "normal".

L'option $S vous permet encore de faire sortir la taille (Size) d'un fichier mais cette taille se résume cependant souvent au nombre d'enregistrements occupés.

STAT <nom de fichier.XXX> $<option> fixe le ou les (avec l'emploi de jokers) fichiers sur l'état <option>. Les options suivantes sont possibles:

$R/O - (Read Only) fixer protection
$R/W - (Read Write) supprimer protection
$SYS - (SYStem) fixer propriété système
$DIR - (DIRectory) supprimer propriété système

STAT <lecteur>:DSK: liste de la façon suivante la table fournie dans la section BIOS:

```
    A: Drive Characteristics
 1368: 128 Byte Record Capacity
  171: Kilobyte Drive Capacity
   64: 32 Byte Directory Entries
   64: Checked Directory Entries
  128: Records/ Extent
    8: Records/ Block
   36: Sectors/ Track
    2: Reserved Tracks
```

STAT VAL: sort une liste des instructions possibles avec STAT qui se présente ainsi:

```
Temp R/O Disk: d:=R/O
Set Indicator: d:filename.typ $R/O $R/W $SYS $DIR
Disk Status : DSK: d:DSK:
User Status : USR:
IObyte Assign:
CON: = TTY: CRT: BAT: UC1:
RDR: = TTY: PTR: UR1: UR2:
PUN: = TTY: PTP: UP1: UP2:
LST: = TTY: CRT: LPT: UL1:
```

Application:

L'instruction STAT donne un peu plus d'informations sur la taille et la répartition des fichiers que ne le fait l'instruction DIR. On peut d'autre part faire sortir l'affectation actuelle des périphériques d'entrée/sortie (DEVices).

Comme l'instruction STAT contient les instructions CP/M 3.0 SET, DEVICE, SHOW et en partie DIR avec options, vous pouvez également lire la signification de ces instructions dans les sections qui leur sont consacrées.

IX.33 SYSGEN (CP/M 2.2)

SYSGEN.COM

Copie le système CP/M 2.2 sur disquette

Format d'entrée:

SYSGEN
SYSGEN FICHIER.XXX
SYSGEN *

Description:

SYSGEN écrit le résultat d'une instruction MOVCPM (voir plus bas) sur la piste système d'une disquette. SYSGEN * écrit sur disquette le système créé immédiatement auparavant par l'instruction MOVCPM. SYSGEN <nom de fichier> sauvegarde un fichier système créé par MOVCPM sur les pistes système d'une disquette. Enfin SYSGEN sans paramètres demande les disquettes source et objet et copie alors ce système.

Avec MOVCPM, on peut décaler CP/M 2.2 dans la mémoire et c'est parfois nécessaire. Vous pouvez décaler CP/M par pas de 256 octets. La position dans la mémoire est indiquée par une valeur entre 64 et 179. La syntaxe est:

MOVCPM <taille>*   Exemple: MOVCPM 178*

MOVCPM 178* produit un CP/M décalé de 256 octets vers le "bas" par rapport au CP/M standard. Vous pouvez maintenant sauvegarder sur disquette ce CP/M ainsi décalé ou le sauvegarder directement comme système avec SYSGEN.

IX.34 TYPE (CP/M 2.2 et CP/M 3.0)

TYPE intégré et TYPE.COM

Affiche un fichier de texte à l'écran

Format d'entrée:

TYPE
TYPE FICHIER.XXX
TYPE AB?.*
TYPE FICHIER.XXX[NO PAGE]
TYPE AB?.*[NO PAGE]

Description:

L'instruction TYPE liste sur l'écran un ou plusieurs fichiers. Dès que l'écran est plein, l'affichage s'arrête pour ne reprendre qu'après que vous ayez appuyé sur la touche espace ou sur ENTER. Si vous entrez l'option "NO PAGE" avec l'instruction, la sortie sur écran ne s'arrêtera pas.

Application:

La version intégrée de TYPE peut être appelée pour n'importe quel lecteur de disquette et pour n'importe quelle zone utilisateur. Lorsque vous voulez interrompre le programme, entrez Control C (^C). Vous pouvez également obtenir que votre fichier soit imprimé en entrant encore l'instruction Control P (^P) avant ENTER.

IX.35 USER (CP/M 2.2 et CP/M 3.0)

USER instruction intégrée

Change l'actuelle zone utilisateur

Format d'entrée:

USER
USER n

Description:

Chaque disquette peut être divisée en 16 zones utilisateur différentes. On peut ainsi avoir une zone utilisateur pour chaque type de travail, dans laquelle ne figureront que les fichiers et programmes nécessaires pour ce type de travail. Les fichiers système de la zone utilisateur zéro peuvent être appelés à partir de toutes les zones utilisateur.

Application:

La zone utilisateur actuelle est indiquée dans l'interrogation CP/M (1B>) et elle peut être modifiée en entrant USER. Si vous n'indiquez pas la nouvelle zone, le programme vous la demandera. Le programme SHOW vous permet de faire afficher les zones utilisateur actives.

IX.36 XREF (CP/M 3.0)

XREF.COM

Crée une liste de référence pour les programmes assembleur

Format d'entrée:

XREF FICHIER
XREF FICHIER $P

Description:

Les assembleurs MAC et RMAC créent une liste alphabétique de tous les symboles et de leurs valeurs respectives dans un programme. XREF rend cette fonction encore plus pratique par le fait qu'il indique en outre dans quelle ligne de programme chaque symbole se trouve. Le programme XREF a besoin des fichiers SYM et PRN.

X. Différences entre les divers ordinateurs CPC

La société Amstrad qui travaille en collaboration avec Locomotive Software peut se vanter d'avoir apporté ces derniers temps une grande animation dans le marché de la micro. Elle a en effet tout de même réussi, en un peu moins d'un an, à mettre 3 (!) ordinateurs sur le marché. Ces ordinateurs étaient ou sont censés être compatibles mais il existe entre eux quelques différences dûes à certaines modifications importantes.

Le premier ordinateur de cette série a été l'Amstrad CPC 464. Cet ordinateur a un lecteur de cassette intégré et il a été et est encore très apprécié, grâce à son Basic rapide et puissant ainsi que grâce à son prix peu élevé. Peu de temps après, la société AMSTRAD annonça une nouvelle machine: le CPC 664.

La différence principale avec le CPC 464 devait être constituée par le lecteur de disquette intégré. Les deux machines ont une mémoire RAM de 64 K octets, dont une part importante est cependant occupée par l'écran (16 K) et les variables système. L'utilisateur dispose librement en Basic de 42249 octets. Le CPC 664 a un Basic légèrement modifié (V 1.1) qui corrige quelques erreurs du Basic du CPC 464 et qui dispose de quelques nouvelles instructions. Pour les deux ordinateurs, CPC 464 et CPC 664, CP/M 2.2 est nécessaire. CP/M 2.2 est la version courante de CP/M pour les ordinateurs de 64 K octets.

Mais il n'y a pas que la vie interne de la machine qui se soit modifiée, l'apparence extérieure a également subi une transformation: le CPC 664 est un peu plus plat et il dispose d'un autre clavier avec un bloc curseur surélevé. Le CPC est devenu plus professionnel. Les maisons de logiciel ont cependant quelques problèmes avec les deux "frères" car la compatibilité annoncée n'est pas réalisée. Cela va d'ailleurs de soi: lorsqu'on élimine des erreurs du Basic et qu'on ajoute des instructions, les deux ordinateurs ne peuvent bien sûr plus être compatibles.

Mais les programmeurs en langage machine sont encore bien plus

durement touchés car c'est ici surtout que des modifications ont été apportées. Mais heureusement: il y a CP/M. Ici rien n'a changé pour l'utilisateur. Vous voyez tout l'intérêt qu'il y a à disposer d'un standard.

Comme les sociétés concurrentes ont mis sur le marché des ordinateurs disposant de 128 K octets et plus, la société AMSTRAD se vit contrainte, avec ses partenaires, de sortir elle aussi un ordinateur 128 K octets. Et c'est ainsi qu'AMSTRAD surprit son monde avec l'AMSTRAD CPC 6128.

Le CPC 6128 dispose également d'un lecteur de disquette intégré et c'est certainement pourquoi le premier chiffre du nom de cet ordinateur est aussi un "6". Le CPC 6128 ressemble aussi plus au CPC 664 qu'au CPC 464. L'ordinateur 128 K octets est encore plus plat, très plat. Il dispose d'un clavier presque encore plus professionnel que celui du CPC 664. Les touches curseur ont été déplacées et ce clavier nécessite un certain temps d'adaptation: la touche COPY n'est plus au milieu du bloc curseur et les touches SHIFT ne se trouvent pas dans la ligne inférieure mais au dessus des touches "CONTROL" et "ENTER". Cela entraîne au moins au début des fautes de frappe de temps à autre.

Le Basic n'a pas été modifié dans la version CPC 6128, la version intégrée est toujours la version 1.1. AMSDOS, c'est-à-dire le système d'exploitation intégré pour le lecteur de disquette est fort heureusement identique pour les trois ordinateurs. Les lecteurs de disquette 3 pouces sont par ailleurs des lecteurs très rapides et très fiables comme en conviendront certainement ceux qui ont l'habitude d'utiliser également d'autres lecteurs de disquette.

Les connexions à l'arrière de l'ordinateur ont également été un peu modifiées. Les connexions du lecteur de disquette et de l'imprimante (Centronics) ont progressé en qualité. Le CPC 6128 dispose en outre également d'une connexion appelée extension.

Lorsque vous achetez le DDI-1 ou le CPC 664, vous recevez une disquette système sur laquelle figurent CP/M 2.2 ainsi que le langage de programmation LOGO. Sur le CPC 6128, ce sont même deux disquettes qui vous sont fournies car tous les fichiers de CP/M 3.0 ne rentraient pas sur une seule disquette. Les textes HELP notamment, c'est-à-dire les textes que vous pouvez appeler avec l'instruction CP/M 3.0 HELP, prennent beaucoup de place sur la disquette. Les acquéreurs d'un CPC 6128 reçoivnt également une disquette comportant CP/M 2.2.

La plus importante différence entre le nouveau CPC et ses prédécesseurs est toutefois la place mémoire disponible. Elle a ici été tout simplement multipliée par deux. Le programmeur Basic en profite cependant moins que le programmeur en langage machine; il ne dispose toujours que de 42249 octets. On ne peut rien y changer sans recourir à des astuces. Mais les 128 K octets de mémoire disponible sur le CPC 6128 permettent à cet ordinateur de faire le saut de CP/M 2.2 à CP/M 3.0. CP/M 3.0 est la version courante de CP/M pour les ordinateurs 128 K octets.

Les deux versions de CP/M permettent de travailler de façon très agréable. Bien entendu, CP/M 3.0 est un peu plus puissant mais cela est dans la nature des choses. C'est à cela que sert le progrès et la société Digital Research développe toujours un peu plus son programme Nobel CP/M.

Comme cet ouvrage doit servir de livre d'entraînement pour tous les ordinateurs CPC, les instructions de CP/M 2.2 sont décrites aussi bien que celles de CP/M 3.0. Presque toutes les instructions CP/M 2.2 peuvent également être utilisées sous CP/M 3.0. Les instructions qui ne peuvent être utilisées que sous CP/M 3.0 sont indiquées spécialement.

ANNEXE 1

TABLE DE CONVERSION

TABLE DE CONVERSION DECIMAL - HEXADECIMAL - BINAIRE

décimal	hexa	binaire
0	&00	&X00000000
1	&01	&X00000001
2	&02	&X00000010
3	&03	&X00000011
4	&04	&X00000100
5	&05	&X00000101
6	&06	&X00000110
7	&07	&X00000111
8	&08	&X00001000
9	&09	&X00001001
10	&0A	&X00001010
11	&0B	&X00001011
12	&0C	&X00001100
13	&0D	&X00001101
14	&0E	&X00001110
15	&0F	&X00001111
16	&10	&X00010000
17	&11	&X00010001
18	&12	&X00010010
19	&13	&X00010011
20	&14	&X00010100
21	&15	&X00010101
22	&16	&X00010110
23	&17	&X00010111
24	&18	&X00011000
25	&19	&X00011001
26	&1A	&X00011010
27	&1B	&X00011011
28	&1C	&X00011100
29	&1D	&X00011101
30	&1E	&X00011110
31	&1F	&X00011111
32	&20	&X00100000
33	&21	&X00100001
34	&22	&X00100010
35	&23	&X00100011
36	&24	&X00100100
37	&25	&X00100101
38	&26	&X00100110
39	&27	&X00100111
40	&28	&X00101000
41	&29	&X00101001
42	&2A	&X00101010
43	&2B	&X00101011
44	&2C	&X00101100
45	&2D	&X00101101
46	&2E	&X00101110
47	&2F	&X00101111
48	&30	&X00110000
49	&31	&X00110001
50	&32	&X00110010
51	&33	&X00110011

TABLE DE CONVERSION DECIMAL - HEXADECIMAL - BINAIRE

décimal	hexa	binaire	décimal	hexa	binaire
52	&34	&X00110100	78	&4E	&X01001110
53	&35	&X00110101	79	&4F	&X01001111
54	&36	&X00110110	80	&50	&X01010000
55	&37	&X00110111	81	&51	&X01010001
56	&38	&X00111000	82	&52	&X01010010
57	&39	&X00111001	83	&53	&X01010011
58	&3A	&X00111010	84	&54	&X01010100
59	&3B	&X00111011	85	&55	&X01010101
60	&3C	&X00111100	86	&56	&X01010110
61	&3D	&X00111101	87	&57	&X01010111
62	&3E	&X00111110	88	&58	&X01011000
63	&3F	&X00111111	89	&59	&X01011001
64	&40	&X01000000	90	&5A	&X01011010
65	&41	&X01000001	91	&5B	&X01011011
66	&42	&X01000010	92	&5C	&X01011100
67	&43	&X01000011	93	&5D	&X01011101
68	&44	&X01000100	94	&5E	&X01011110
69	&45	&X01000101	95	&5F	&X01011111
70	&46	&X01000110	96	&60	&X01100000
71	&47	&X01000111	97	&61	&X01100001
72	&48	&X01001000	98	&62	&X01100010
73	&49	&X01001001	99	&63	&X01100011
74	&4A	&X01001010	100	&64	&X01100100
75	&4B	&X01001011	101	&65	&X01100101
76	&4C	&X01001100	102	&66	&X01100110
77	&4D	&X01001101	103	&67	&X01100111

TABLE DE CONVERSION DECIMAL - HEXADECIMAL - BINAIRE

décimal	hexa	binaire	décimal	hexa	binaire
104	&68	&X01101000	130	&82	&X10000010
105	&69	&X01101001	131	&83	&X10000011
106	&6A	&X01101010	132	&84	&X10000100
107	&6B	&X01101011	133	&85	&X10000101
108	&6C	&X01101100	134	&86	&X10000110
109	&6D	&X01101101	135	&87	&X10000111
110	&6E	&X01101110	136	&88	&X10001000
111	&6F	&X01101111	137	&89	&X10001001
112	&70	&X01110000	138	&8A	&X10001010
113	&71	&X01110001	139	&8B	&X10001011
114	&72	&X01110010	140	&8C	&X10001100
115	&73	&X01110011	141	&8D	&X10001101
116	&74	&X01110100	142	&8E	&X10001110
117	&75	&X01110101	143	&8F	&X10001111
118	&76	&X01110110	144	&90	&X10010000
119	&77	&X01110111	145	&91	&X10010001
120	&78	&X01111000	146	&92	&X10010010
121	&79	&X01111001	147	&93	&X10010011
122	&7A	&X01111010	148	&94	&X10010100
123	&7B	&X01111011	149	&95	&X10010101
124	&7C	&X01111100	150	&96	&X10010110
125	&7D	&X01111101	151	&97	&X10010111
126	&7E	&X01111110	152	&98	&X10011000
127	&7F	&X01111111	153	&99	&X10011001
128	&80	&X10000000	154	&9A	&X10011010
129	&81	&X10000001	155	&9B	&X10011011

TABLE DE CONVERSION DECIMAL - HEXADECIMAL - BINAIRE

décimal	hexa	binaire	décimal	hexa	binaire
156	&9C	&X10011100	182	&B6	&X10110110
157	&9D	&X10011101	183	&B7	&X10110111
158	&9E	&X10011110	184	&B8	&X10111000
159	&9F	&X10011111	185	&B9	&X10111001
160	&A0	&X10100000	186	&BA	&X10111010
161	&A1	&X10100001	187	&BB	&X10111011
162	&A2	&X10100010	188	&BC	&X10111100
163	&A3	&X10100011	189	&BD	&X10111101
164	&A4	&X10100100	190	&BE	&X10111110
165	&A5	&X10100101	191	&BF	&X10111111
166	&A6	&X10100110	192	&C0	&X11000000
167	&A7	&X10100111	193	&C1	&X11000001
168	&A8	&X10101000	194	&C2	&X11000010
169	&A9	&X10101001	195	&C3	&X11000011
170	&AA	&X10101010	196	&C4	&X11000100
171	&AB	&X10101011	197	&C5	&X11000101
172	&AC	&X10101100	198	&C6	&X11000110
173	&AD	&X10101101	199	&C7	&X11000111
174	&AE	&X10101110	200	&C8	&X11001000
175	&AF	&X10101111	201	&C9	&X11001001
176	&B0	&X10110000	202	&CA	&X11001010
177	&B1	&X10110001	203	&CB	&X11001011
178	&B2	&X10110010	204	&CC	&X11001100
179	&B3	&X10110011	205	&CD	&X11001101
180	&B4	&X10110100	206	&CE	&X11001110
181	&B5	&X10110101	207	&CF	&X11001111

TABLE DE CONVERSION DECIMAL - HEXADECIMAL - BINAIRE

décimal	hexa	binaire	décimal	hexa	binaire
208	&D0	&X11010000	234	&EA	&X11101010
209	&D1	&X11010001	235	&EB	&X11101011
210	&D2	&X11010010	236	&EC	&X11101100
211	&D3	&X11010011	237	&ED	&X11101101
212	&D4	&X11010100	238	&EE	&X11101110
213	&D5	&X11010101	239	&EF	&X11101111
214	&D6	&X11010110	240	&F0	&X11110000
215	&D7	&X11010111	241	&F1	&X11110001
216	&D8	&X11011000	242	&F2	&X11110010
217	&D9	&X11011001	243	&F3	&X11110011
218	&DA	&X11011010	244	&F4	&X11110100
219	&DB	&X11011011	245	&F5	&X11110101
220	&DC	&X11011100	246	&F6	&X11110110
221	&DD	&X11011101	247	&F7	&X11110111
222	&DE	&X11011110	248	&F8	&X11111000
223	&DF	&X11011111	249	&F9	&X11111001
224	&E0	&X11100000	250	&FA	&X11111010
225	&E1	&X11100001	251	&FB	&X11111011
226	&E2	&X11100010	252	&FC	&X11111100
227	&E3	&X11100011	253	&FD	&X11111101
228	&E4	&X11100100	254	&FE	&X11111110
229	&E5	&X11100101	255	&FF	&X11111111
230	&E6	&X11100110			
231	&E7	&X11100111			
232	&E8	&X11101000			
233	&E9	&X11101001			

ANNEXE 2

LES CARACTERES DE CONTROLE DE CP/M

Les caractères de contrôle de CP/M

Instruction	Effet
===========	=====
^A	Curseur d'un caractère vers la gauche *
^B	Curseur en début de ligne ou en fin de ligne s'il est déjà en début de ligne
^C	Interrompre CP/M; exécuter un Reset
^E	Curseur sur la ligne suivante
^F	Curseur d'un caractère vers la droite *
^G	Supprimer caractère sous le curseur
^H	Supprimer caractère à gauche du curseur
^I	Curseur sur la prochaine tabulation
^J	Instruction d'exécution
^K	Supprimer du curseur à la fin de la ligne
^M	Comme ENTER
^P	Activer/désactiver l'imprimante
^Q	Activer scrolling après ^S
^R	Répéter ligne d'instructions
^S	Arrêter sortie sur écran
^U	Supprimer huit caractères dans la ligne

^W	Afficher à nouveau ancienne ligne d'instructions *
^X	Supprimer caractère à gauche du curseur
^Z	Fin de chaîne de caractères pour PIP et ED

* signifie: disponible uniquement pour CP/M avec "bank switching"

ANNEXE 3

TOUS LES PARAMETRES PIP

Paramètres PIP

A	Archiv	Ne copie que les fichiers qui ont été créés ou modifiés depuis le dernier archivage. La fonction tampon de date et heure doit avoir été fixée.
C	Confirm	CP/M demande pour chaque fichier si ce fichier peut être copié.
Dn	Delete	Suppression de n colonnes. PIP supprime tous les caractères qui figurent dans le fichier après la n-ième colonne. N'utiliser que pour des fichiers de texte.
E	Echo	Affiche texte sur l'écran. Le contenu des fichiers en train d'être copiés est affiché sur l'écran. Ne pas utiliser en même temps que le paramètre "N". N'utiliser que pour les fichiers de texte.
F		Interdire le caractère pour "page suivante" (form feed). Quelques imprimantes ont besoin d'un (^L) pour avancer à la page suivante. Si votre imprimante n'en a pas besoin, vous pouvez empêcher les caractères de commande correspondants avec "F".
Gn	Get	Amener un fichier de ou vers la zone utilisateur "n". Cette indication doit figurer directement à la suite du premier nom de fichier et c'est la seule qui puisse figurer à cet emplacement. Exemple: PIP B:[G5]=TEXTE.TXT[G1]

Copiera le fichier TESTE.TXT de la zone utilisateur "1" vers la zone utilisateur "5".

H Hex — Transfert de données hexadécimales. Vous devriez toujours entrer cette option lorsque vous voulez transférer des fichiers HEX. PIP teste alors le contenu pour voir s'il est écrit dans un format INTEL correct.

I Ignore — Ignorer le caractère fin de fichier (EOF) dans les fichiers HEX. Entrez cette option pour chaque fichier, sauf pour le dernier. L'option "H" doit être fixée en même temps que l'option "I". Utilisez pour le dernier fichier le paramètre "H":

PIPFICHIER.HEX=PROG1.HEX[I],PROG2.HEX[H]

K — Interdire l'affichage du nom de fichier lors de la copie.

L — Convertir toutes les majuscules en minuscules. N'utilisez cette option que pour les fichiers de texte et utilisez en plus le paramètre "Z" pour les fichiers WordStar.

N — Numérotage continu des lignes d'un fichier. Les numéros commencent à 1 pour la première ligne et sont augmentés de 1 par ligne. Les nombres ne peuvent comporter plus de 6 chiffres, les chiffres non-utilisés apparaissent comme espaces. Le nombre est suivi d'un double point et d'un espace. Ne pas utiliser en même temps que les options "E" ou "N". Pour les fichiers WordStar, entrer en plus l'option "Z".

N2	Numérote les lignes pour un programme BASIC.
O Object	Transfert de fichiers objet. Est utilisé pour transférer les fichiers non-texte et non-COM. Ne pas utiliser pour les fichiers de texte.
Pn Page	Fixe la longueur de page. La valeur pré-fixée est de 60 lignes par page. Vous devriez entrer en même temps l'option "F" pour que les caractères pour fin de page (^L) soient supprimés. N'utiliser que pour les fichiers de texte.
Qxxxx^Z	Fin de la copie après cette chaîne de caractères. PIP copie un fichier jusqu'à la chaîne de caractères entrée inclusivement. Utilisez l'instruction avec deux lignes d'instructions pour que la chaîne de caractères ne soit pas convertie en majuscules: PIP CON:=TEXTE.TXT[Astuce] Si vous écrivez tout dans la première ligne à la suite de PIP, l'entrée apparaîtra en majuscules.
R	Copie les fichiers système. Ces fichiers ne sont pas affichés par DIR et ne sont normalement pas copiés par PIP. "R" permet de lever cette limitation.
Sxxxx^Z	Commencer à cette chaîne de caractères. PIP commence la copie à cette chaîne de caractères. Ecrivez cette chaîne de caractères sur deux lignes, sinon la chaîne de caractères sera convertie en majuscules. N'utiliser que pour les fichiers de texte.

Tn	Tab	Fixer l'espacement du tabulateur. CP/M travaille normalement avec un pas de huit caractères. Cela ne marche cependant pas lors de l'impression d'un fichier et le pas doit être défini. "n" indique le nombre d'espaces pour un pas de tabulation. N'utiliser que pour les fichiers de texte.
U		Conversion en majuscules. Convertit toutes les minuscules en majuscules. N'utiliser que pour les fichiers de texte. Pour les fichiers WordStar, utiliser en plus l'option "Z".
V	Verify	Ce paramètre fait que PIP comparera précisément le fichier copié avec le fichier source.
W	Write	Effacer même les fichiers comportant l'attribut (RO). Ces fichiers ne peuvent normalement qu'être lus mais non modifiés ou supprimés. Avec l'option "W", ces fichiers seront supprimés sans demande de confirmation.
Z	Zero	Supprimer le bit de parité. Le jeu de caractères ASCII n'utilise que sept bits. Le huitième bit est utilisé par différents programmes pour des tâches différentes. La fonction "Z" n'est pas nécessaire lors de la copie vers des unités ASCII telles que CON: ou LST:.

ANNEXE 4

LES PARAMETRES SET

OPTION SIGNIFICATION
====================

DIR rend un fichier système à nouveau visible dans le catalogue normal

SYS fait d'un fichier un fichier système

RO fait qu'un fichier ne peut être que lu

RW fait qu'un fichier peut être lu et modifié

ARCHIV=OFF fixe l'attribut ARCHIV sur "non". Cela signifie que ce fichier n'a pas encore été sauvegardé (archivé). Le programme PIP avec l'option [A] peut copier des fichiers avec l'attribut ARCHIV=OFF. Vous entrez l'instruction PIP avec l'étoile pour les noms de fichier et PIP copiera tous les fichiers qui ont été modifiés depuis la dernière opération de copie avec PIP et depuis l'option [A]. Après que PIP ait effectué la copie, il fixe les attributs de fichier sur ARCHIV=ON

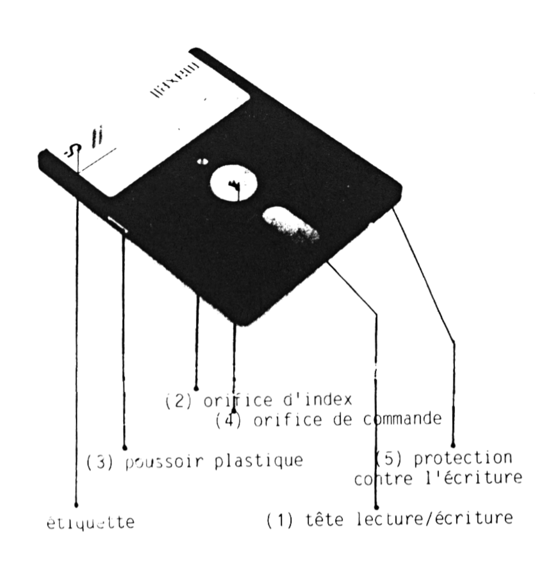
ARCHIV=ON fixe l'attribut ARCHIV sur "oui". Cela signifie que ce fichier a été sauvegardé. Normalement, PIP, avec l'option [A], modifie cet attribut après la sauvegarde de fichiers. Vous pouvez également modifier vous-même l'attribut si vous utilisez l'instruction SET.

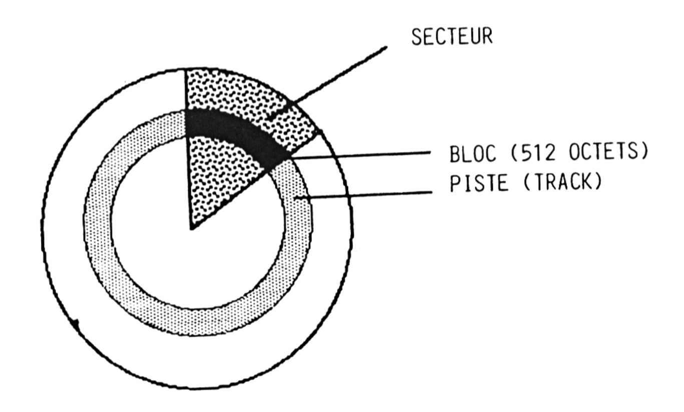
F1=ON/OFF active ou désactive l'attribut de fichier F1 qui est défini par l'utilisateur.

F2=ON/OFF active ou désactive l'attribut de fichier F2 qui est défini par l'utilisateur.

F3=ON/OFF active ou désactive l'attribut de fichier F3 qui est défini par l'utilisateur.

F4=ON/OFF active ou désactive l'attribut de fichier F4 qui est défini par l'utilisateur.


(2) orifice d'index
(4) orifice de commande
(3) poussoir plastique
(5) protection contre l'écriture
étiquette
(1) tête lecture/écriture


SECTEUR
BLOC (512 OCTETS)
PISTE (TRACK)

Attention

MICRO APPLICATION lance sa nouvelle gamme entièrement en français de logiciel utilitaire et professionnel.

Demandez notre catalogue gratuit :

NOM : ....

Prénom : ....

Adresse : ....

Code Postal : ....

Achevé d'imprimer en octobre 1985
sur les presses de l'imprimerie Laballery et Cie
58500 Clamecy
Dépôt légal : octobre 1985
Numéro d'imprimeur : 510013

IMPRIMÉ EN FRANCE

[redacted]

Enfin un livre qui vous permet de maîtriser CP/M sur les CPC 464, 664 et 6128 ! Tout utilisateur d'un CPC trouvera dans cet ouvrage l'aide et les explications nécessaires à une bonne utilisation et compréhension de CP/M comme par exemple le stockage des données, la protection contre l'écriture, la codification ASCII, l'utilisation des programmes utilitaires CP/M et même « l'intérieur » de CP/M pour les programmeurs avancés. Ce livre couvre les versions CP/M2.2 et CP/MPLUS (3.0) pour les AMSTRAD CPC 464, 664 & et bien entendu le tout nouveau CPC 6128.

Les sujets traités couvrent notamment :

— Les fonctions de CP/M
— La disquette système
— Les règles pour les noms de fichier
— Les instructions intégrées USER, DIR, ERASE
— Les instructions transitoires SET, PROTECT, SHOW, SUBMIT
— Tout sur PIP
— Imprimer plusieurs fichiers en continue
— Lire différents formats de disquettes
— Les différences entre les différents CPC
— et beaucoup d'autres choses...

Ce livre a été écrit par un spécialiste de CP/M auteur de différents ouvrages de référence, A. WEILER et J. SCHIEB qui est un expert du CPC et a collaboré notamment au célèbre Livre du Lecteur de Disquette AMSTRAD.

Prix : 149 F TTC
Réf. : ML 128

ISBN : 2-86899-025-8


AMSTRAD
LE LIVRE DU CP/M2.2
ET CP/M PLUS (3.0)
12



Document numérisé avec amour par

AMSTRAD CPC

MÉMOIRE ÉCRITE



https://acpc.me/